

(10) International Publication Number
WO 01/77883 A1

- For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.*

FIG. 1 is a block diagram of a system architecture for creating and updating Active Server Pages (ASP) via the Internet. The system includes a client (10) with a browser (20) and a server (14) with a portal engine (26) and a portal database (12). The client sends a request (16) to the server, which then sends a response (17) back. The server also receives information (22) from the Internet and sends it (24) to the portal engine. The portal engine then sends information (28) to the portal database. The portal database is also connected to a network (12).

(57) Abstract: The present invention relates to web sites, databases and search engines, and, in particular, relates to a combined web site and database which is searchable using an improved search engine. More particularly, the present invention relates to a method of creating vertical portals or search infrastructures, and integrating them into existing web sites. The invention accordingly provides for a web page including displayed information wherein at least some of the displayed information corresponds with stored information which is stored in fields in tables in a web page database, wherein the displayed information includes a plurality of displayed information units, the stored information includes a plurality of stored information units, and each displayed information unit is displayed in a displaying region on the web page, and wherein the web page database includes a primary web page table including a primary web page field which contains a primary web page database identifier which uniquely identifies the web page database. The invention also provides for a web site comprising a plurality of web pages, and a set of affiliate web sites and corresponding databases, including a super database made up of a series of portals each including their own affiliated set of web site databases.

System and method for creating and searching web sites

Field of the invention

The present invention relates to web sites, databases and search engines, and, in particular, relates to a combined web site and database which is searchable using an improved search engine. More particularly, the present invention relates to a method of creating vertical portals or search infrastructures, and integrating them into existing web sites.

Background of the invention

The Internet is growing at an alarming rate. Its use is becoming increasingly business focussed. Unfortunately, the differentiation between general information (general content) and business information (business/transactional content) is not clear enough when searches are performed using traditional means.

At present, known search engines act as individual entities, so finding information becomes a combination of a consumer or business going to various typically non-independent sites, and hopefully coming across what they are looking for.

Traditional business specific directories are limited in the way information can be found and maintained.

The present invention was developed for a number of reasons, the main ones being:

- a) to standardise certain content on the Internet and provide a unique search engine architecture which can be used to create and locate that content;
- b) to provide an alternative to current web hosting methods, which offers cost benefits, and ease of maintenance; and
- c) to provide a transparent solution for businesses who would like (content specific) search engine functionality integrated into their existing web sites.

The present invention centres around the process of creating search engine infrastructure for third parties.

The present invention is an improvement over the prior art because its hierarchical architecture allows for third party/independent, sub-portals (search engines) to be created that contain specific information. These search engines might contain restaurant information, accounting information, or health information.

These sub-portals' data can then be amalgamated into a higher level portal, which may be industry specific.

The data within these sub-portals can be maintained through a maintenance interface. New content can be added, activated, deactivated or changed via a user name and password.

Additional advantages of the present invention can be summarised as follows:

- a) A third party vertical portal structure can be implemented ready for update by the client within hours. The architecture of the present invention is open, so that programming is not required. Users need only choose the web site structure they like and fill in the details in the information entry interface.

b) The search engine (SQL server side) aspect of the present invention contains all the business rules for the way the invention operates and functions. If a change is required, it can be implemented in the back end, which will effectively replicate the functionality change to all vertical portal sites:

c) The present invention's back end architecture has been implemented to maximise the performance of interactions between the interface and the database content.

The present invention is capable of creating at least the following structures:

1. a single web site for a single business (say a 10 page web site);
2. a collection of web sites which fall under a search structure (accessed by a third party web site) and are visible in the super database/search engine; and
3. a collection of web sites which fall under a search structure (accessed by a third party web site) which are invisible in the super database/search engine.

Summary of the invention

Various exemplary, non-limiting aspects of the present invention are defined and disclosed in the following paragraphs:

According to one aspect of the invention there is provided a web page including displayed information wherein at least some of the displayed information corresponds with stored information which is stored in fields in tables in a web page database, wherein the displayed information includes a plurality of displayed information units, the stored information includes a plurality of stored information units, and each displayed information unit is displayed in a displaying region on the web page, and wherein the web page database includes a primary web page table including a primary web page field which contains a primary web page database identifier which uniquely identifies the web page database.

In a preferred form of the invention, each displaying region corresponds to a field in the web page database such that the displayed information unit (DIU) displayed in that region corresponds with the stored information unit (SIU) stored in its corresponding field in the database.

Conveniently, the position, size, colour and/or orientation of the display regions on the page are defined by a page layout.

Typically, the fields store SIUs chosen from a group including:

- a) characters;
- b) numbers;
- c) pictures;
- d) links; and
- e) dates.

The SIUs may be chosen from a group including:

- a) names;
- b) addresses;
- c) phone numbers;
- d) fax numbers;
- e) email addresses;
- f) industry categories;
- g) product names;
- h) product descriptions;
- i) service names;
- j) service descriptions;
- k) announcements;
- l) catalogues;
- m) menus;
- n) keywords;
- o) event names;
- p) event descriptions;
- q) event dates;
- r) pictures; and
- s) banner advertisements.

Typically, the display regions display DIUs are chosen from the group including:

- a) characters;
- b) numbers;
- c) pictures;
- d) links.

The display regions may be chosen from a group including:

- a) name regions;
- b) address regions;
- c) phone number regions;
- d) fax number regions;
- e) email address regions;
- f) industry category regions;
- g) product name regions;
- h) product description regions;
- i) service name regions;
- j) service description regions;
- k) announcement regions;
- l) catalogue regions;

- m) menu regions;
- n) keyword regions;
- o) event name regions;
- p) event description regions; or
- q) event date regions;
- r) owner contact information regions;
- s) picture regions;
- t) product information regions;
- u) announcement information regions;
- v) link information regions; and
- w) menu information regions; and
- x) banner advertisement regions.

Typically, all of the displayed information corresponds with stored information which is stored in the fields in the tables in the web page database.

Advantageously, the primary web page table is linked to at least one subsidiary web page table by a foreign key, with the at least one subsidiary web page table conveniently being linked to further subsidiary tables by further foreign keys.

Preferably, each web page references, or is referenced by, at least one other web page via a link.

The invention extends to a web site including a plurality of web pages, wherein each web page includes displayed information where at least some of the displayed information corresponds with stored information which is stored in fields in tables in a web page database, wherein the displayed information includes a plurality of displayed information units, the stored information includes a plurality of stored information units, and each displayed information unit is displayed in a displaying region on the web page, and, wherein the web site has a unique web site identifier which corresponds to a unique primary web site database identifier such that the displayed information on a particular web site is stored in its corresponding web site database, and wherein each web page references, or is referenced by, at least one other web page via a link.

Conveniently, the displaying regions on the web pages are arranged according to a plurality of page layouts.

The types of page layouts may include:

- a) home page layouts;
- b) menu page layouts;
- c) newsletter page layouts;
- d) announcement page layouts;
- e) product page layouts;
- f) service page layouts;
- g) link page layouts;

- h) general information page layouts; and
- i) content page layouts

Advantageously, the web site has at least one relational identifier, with the relational identifier of a particular web site being stored in its corresponding web site database.

5 Typically, the at least one relational identifier is chosen from the group including:

- a) an affiliate identifier;
- b) a parent identifier;
- c) a child identifier;
- d) a vertical portal identifier; or
- 10 e) a horizontal portal identifier.

The invention extends further to a set of affiliate web sites including a plurality of web sites of the type described above wherein each web site has the same affiliate identifier.

The invention further provides an hierarchical set of affiliate web sites including a plurality of affiliate web sites of the type described above wherein a parent web site includes at least one child identifier and a child web site includes a parent identifier, such that the child identifier of the parent web site corresponds to the parent identifier of the child web site.

According to a further aspect of the invention there is provided a computer implemented method of forming the web page described above by:

- a) defining a database including at least one table including at least one field; and
- 20 b) associating at least one of the fields with at least one display region on the web page.

The computer implemented method extends to the formation of a plurality of web pages of a type described above by:

- a) creating a template web page including a plurality of display regions whose size, orientation and/or layout is defined by a template page layout;
- 25 b) creating a template database including at least one table including a plurality of fields containing default field information and being associated with one or more display regions on the template web page;
- c) copying the template database to create a new database having substantially the same table attributes, field attributes and default field information as the template database;
- 30 d) creating a new primary database identifier and storing it in a primary field of a primary table of the new database.

- e) copying the template web page to create a new web page having a unique web site identifier which corresponds with the new primary database identifier and having a plurality of display regions whose size, orientation and/or layout matches the template page layout;
- f) associating the fields of the new database with the display regions on the new web page; and
- 5 g) repeating steps b) to d) until the desired number of new web pages is reached.

Advantageously, the step of creating a template web page includes the step of accepting layout specifications from a user and applying those specifications to the web page.

The invention further provides a computer implemented method of forming a web site of the type described above by forming a first web page in accordance with the method defined above, forming a second web
10 page in accordance with the method defined above, adding a link, which references the second web page to the first web page, and preferably adding a link which references the first web page to the second web page.

The invention still further provides a computer implemented method of creating and adding a new web page to an web site, wherein the existing web site is a web site as defined above, by forming a new web page in accordance with the method described above, adding a first new link, which references the new web page to a first
15 existing web page in the existing web site, and preferably adding a second new link, which references a new web page to a second existing web page.

The method may extend to include the additional step of adding a third new link, which references an existing web page to the new web page.

There is further provided a computer implemented method of forming a set of affiliate web sites of the type described above by forming a first web site using the method described above, forming a second web site
20 using the same method defined above, and assigning the same affiliate identifier to the first and second web sites.

There is still further provided a computer implemented method of forming an hierarchical set of affiliate web sites of the type described above by forming a set of affiliate web sites using the method defined above, assigning a child identifier to the parent web site, and assigning a matching parent identifier to the child web site.

The invention extends to a data entry interface for updating displaying information on a web page of the type described above including a plurality of information entry fields, each of which corresponds to a field in a table in the database.

The interface preferably includes an uploader for uploading entered information from the information entry fields into the stored information fields in the database and may also include a downloader for downloading
30 stored information from the stored information fields in the database into the information entry fields.

The interface may further include a downloader for downloading stored information from the stored information fields in the database into the information entry fields.

The entered information may include a plurality of entered information units (EIUs).

The plurality of information entry fields may be displayed on a plurality of information entry pages, which may be internet web pages, with at least some of the information entry pages preferably being linked to one another.

The information entry pages may include:

- a) personal information entry pages;
- b) product information entry pages;
- c) service information entry pages;
- d) announcement information entry pages;
- e) catalogue information entry pages;
- f) menu information entry pages;
- g) general information entry pages;
- h) additional links information entry pages; and/or
- i) homepage information entry pages.

The information entry pages may further include action choice buttons which may in the form of links or pictures which behave like buttons.

The action choice buttons may be chosen from the group including:

- a) edit buttons;
- b) add buttons;
- c) save buttons; and
- d) activate pick lists.

Pressing the edit button on an information entry page may cause the downloader to download the SIUs from the database fields into the corresponding information entry fields in the information page for editing by the user.

Pressing the add button on an information entry page typically causes the downloader to download the SIUs from the database fields into the corresponding information entry fields in the information page in read only format, so that the user can only add entered information to empty information entry fields.

Pressing the save button on an information entry page typically causes the uploader to upload the entered information units (EIUs) entered by user into the corresponding database fields as SIUs.

Conveniently, selection of the activate option in the activate picklist on an information entry page causes the uploader to change an active indicator in a field in the database to an active state.

According to a still further aspect of the invention there is provided a computer implemented method of generating a search string for use in searching a database having a plurality of database tables each of which has a database table name, wherein each table contains a plurality of database fields each of which has a database field

name and contains stored information, wherein said search string is based upon at least one search parameter, each search parameter having a corresponding database field and wherein said method includes the steps of:

- a) generating a database field selection statement,
- b) generating a database table selection statement:
- c) generating a condition statement,
- d) concatenating said database field selection statement, said database table selection statement and said condition statement to form a search string.

The method preferably includes the additional step of generating a result order statement and concatenating said result order statement to said search string.

Advantageously, the step of generating a database field selection statement is achieved by:

- a) declaring a database field selection string variable;
- b) initialising said string variable;
- c) adding a database field selection command to the start of said string;
- d) adding at least one database field name to said string, followed by a separator;
- e) repeating step d) until the field selection string variable includes all the required database field names; and
- f) removing the final separator in the database field selection string.

The database field selection command may be the SQL selection command "SELECT", and the separator may be a comma.

Conveniently, the step of generating a database table selection statement is achieved by:

- a) declaring a database table selection string variable;
- b) initialising said string variable;
- c) adding a database table selection command to the start of said string;
- d) adding at least one database table name to said string, followed by a separator;
- e) repeating step d) until the table selection string variable includes all the required database table names; and
- f) removing the final separator in said database table selection string.

Typically, the database field selection command is the SQL selection command "FROM".

Preferably, the step of generating a condition statement is achieved by:

- a) declaring a condition string variable;
- b) initialising said string variable;
- c) adding a condition command to the start of said string;
- d) adding at least equation to said string, followed by a separator;
- e) repeating step d) until the condition string variable includes all the required equations; and
- f) removing the final separator from the string.

The condition command may be the SQL selection command "WHERE".

Typically, the equation takes the form of: "database field name = corresponding search parameter".

Preferably, the step of generating a result order statement is achieved by:

- a) declaring a result order string variable;
- b) initialising said string variable.
- c) adding a result order command to the start of said string;
- d) adding at least one database field name to said string, followed by a separator;
- e) repeating step d) until the result order string variable includes all the required database field names; and
- f) removing the final separator from the string.

The result order command may be the SQL result order command "ORDER BY".

The invention still further provides a search string generated by the computer implemented method described above.

The invention also extends to a computer implemented search string generator which generates search strings in accordance with the method described above.

According to a still further aspect of the invention there is provided a web page database including stored information wherein at least some of the stored information corresponds with displayed information which is displayed in displaying regions on a web page, wherein the web page is a web page as defined in any one of the paragraphs above.

There is further provided a web site database including plurality of web page databases of the type described above.

The web site database may include a plurality of web sites of the type described in any of the relevant paragraphs above.

Conveniently, at least some of the stored information in the web site database corresponds with displayed information displayed on the web site of the type described above.

The invention further extends to a super database including a plurality of web site databases of the type described above, with each of the web site databases including an affiliate field which contains an affiliate identifier, wherein web sites which are affiliated with one another preferably have the same affiliate identifier.

Conveniently, each web site database further includes a visibility field containing a visibility variable which has a positive status and a negative status, indicating whether or not the web site database is visible in the super database.

The super database may include a super affiliate field which contains a super affiliate code.

According to a yet further aspect of the invention there is provided an internet searching system including:

a search interface including at least one search field for receiving user supplied search parameters;

a super database of a type described above;

a search engine;

a search results displaying interface for displaying a plurality of search result units;

- 5 wherein said search interface receives said user supplied search parameters and communicates them to said search engine;

said search engine receives said search parameters, generates a query based thereon, searches said super database using said query, produces said plurality of search result units and communicates same to said search results displaying interface; and

- 10 wherein said search results displaying interface receives and displays said search result units.

Typically, the search interface has a plurality of search interface types, and each of said search interface types has an associated search interface identifier.

Conveniently, said search interface communicates its search interface identifier to said search engine along with said user supplied search parameters.

- 15 The search engine typically generates an amended query based upon both said search parameters and said search interface identifier.

Advantageously, the search engine searches said super database using said amended query, producing a plurality of filtered search result units.

- 20 Typically, said filtered search result units include stored information which is relevant to said search interface type.

Advantageously, said search interface type is a super search interface having a super search interface identifier.

The amended query may include a condition that the only web site databases which are searched are those having visibility variables with a positive status.

- 25 The search interface may be an affiliate search interface having an affiliate search interface identifier.

The amended query may include a condition that the only web site databases which are searched are those having an affiliate identifier which corresponds with said affiliate search interface identifier.

The search results displaying interface may have a plurality of interface types, with each of the displaying interface types having an associated displaying interface layout.

- 30 The displaying interface layout typically determines the presentation, colour and arrangement of said search result units on said displaying interface.

Advantageously, the search result units are displayed in a search results table having column headings on said displaying interface and wherein said displaying interface layout determines the presentation, colour and arrangement of said results table.

The displaying interface layout conveniently determine the information displayed in the column headings of the results table.

The displaying interface layout may further determine the colour, size, orientation and appearance of logos and advertising banners on said displaying interface.

The displaying interface type may be determined by the search interface type used, and the search interface type may be a super search interface and the displaying interface type may be a super search results displaying interface.

The search interface type may be an affiliate search interface and the displaying interface type may be an affiliate search results displaying interface. The invention further provides an internet searching system of the type described above wherein the query is a search string of the type described above.

The search results displaying interface may also include at least one search field to enable a user to refine their search.

The invention extends to a computer program product comprising a computer readable medium having thereon computer program code means, when said program is loaded, to make the computer execute procedure to provide a data entry interface of the type described above.

The invention still further provides a server hosting a web page database of a type described above, as well as a server hosting web site database of the type described above and a super database of the type described above.

For purposes of clarity, it should be understood that the terms "affiliate" and "portal" may be used interchangeably.

Brief description of the drawings

Figure 1 shows a schematic high level diagram of a web portal creation system of the invention;

Figure 2 shows a flowchart illustrating the various steps involved in creating a portal;

Figures 3A and 3B show a typical template page layouts for enabling the capture of portal creation information;

Figure 4 shows a typical page layout for creating a product web page;

Figure 5 shows a schematic diagram illustrating a web site and portal updating process of the invention;

Figure 6 shows a typical page layout providing a user with a business updating interface;

Figure 7 shows a typical homepage constructed using the updated information present in the page layout of Figure 6;

Figure 8 shows a schematic diagram illustrating a searching procedure for searching portals created by the portal creation process of the invention:

Figure 9 shows a page layout of a search result; and

Figure 10 shows an entity relationship diagram illustrating the architecture of the portal database.

5 Detailed description of the preferred embodiments

Referring first to Figure 1, a web site portal creation system 10 of the invention has at its heart a portal database 12 residing in a web server 14 and including a database or portal query engine 15. A remote subscriber terminal 16 is linked via the internet to the server web site and is provided with a web browser 18 for enabling portal set up requirements to be captured from the subscriber. A series of data entry template pages 20 are used to capture the portal set up parameters from the subscriber, who provides information such as the number of web sites contained in a particular portal or search infrastructure and the number and type of pages contained within each web site. The information entered into the data entry data pages 20 is transmitted via the internet 22 to the web server 14 in the form of active server pages (ASPs) 24. The portal database 12 includes a series of portals A, B...Z which are created by a series of SQL stored procedures that populate corresponding tables within the portal database 12.

The ASPs are used in the creation of web sites containing various pages, including a web site homepage, a general content page, announcement pages, catalogue pages, cuisine or menu pages, special deal pages and external links. Each web site homepage has a keyword field associated with it for use during location functions, and is created visually via portal creation parameters 26 which are transmitted via the portal engine to the database, which contains within its tables content and visual information.

The database engine 15 also has a number of different enquiry and display functions. These include the acceptance of search parameters and the performance of a search against such parameters. Catalogue parameters are accepted, and catalogue tables are searched, as are announcement parameters and searching of the corresponding announcement table. The same is applicable in respect of menu parameters and product and service keyword parameters.

Further functions include the display of one or more web sites within an individual portal through a URL call. The database engine requests and accepts a user name and password for update access. On successful login, a browser interface is presented for enabling on-line changes.

A request portal function of the database engine captures portal requirement requests, and includes a series of tables recording captured request information which remains in the tables until processed by a request processor procedure. This is run periodically, and creates portal web sites based on the information in the tables in the manner described further on in the specification.

A series of administration update procedures update the portal records, from information passed from the browser-based user maintenance pages. These records correspond with the web site pages. A picklist populator function populates portal-specific picklists when performing updates to the system, and a page process function

runs periodically either to activate or deactivate announcement pages. A fax/email process is run periodically by the database engine so as to fax or email messages contained within the fax or email table.

Turning now to Figure 2, the portal creation process will be described in more detail. Typically, a subscriber or portal creator 34 would belong to an umbrella organisation such as a club or institute, a local chamber of commerce, a shopping centre, or exhibition or conference organisation. The user accesses the portal database via a URL, and the browser 17 displays in sequence a number of ASP pages, including a main page template 24.1, a content page template 24.2, an announcement page template 24.3, a catalogue page template 24.4, a menu page template 24.5 and a deal page template 24.6. The number of required web sites are entered on the main page template 24.1, and the number of pages required in respect of the content, announcement, catalogue, menu and deal pages are similarly entered on the corresponding page templates.

Once the user has entered the portal details, a portal creation confirmation page appears containing the following text:

Are you sure you want to create ten web sites each with:

- 5 product/services
- 5 announcements
- 2 catalogues
- 0 menus
- 0 menu deals?

If the user is satisfied with the portal configuration, the "yes" or create portal button 36 is pressed. This causes the portal creation parameters to be transmitted to the portal web site. Various portal link creation procedures 40 are initiated. These include the initialisation of portal request tables within the database, and the allocation of a unique identification code for the particular portal, as shown at 40.1. The request tables within the portal database are subsequently populated with portal requirement data, including the number of web sites and the number of pages within each web site, as is shown at 40.2.

The portal web site creation procedure is illustrated in more detail at 42. Once the request tables have been populated with the overall number of web sites and the overall number of pages within each web site, each web site is then individually constructed by creating a main page and thereafter the requisite number of content, announce, catalogue, cuisine or menu and deal pages, as shown at 42.1 to 42.6 respectively. The unique web site user name and password is subsequently generated at 42.7 using the first of those letters of the portal code together with the web site unique identifier. The password is created using a time and date stamp seed. Once all of the web sites have been generated, a fax/email notification record is created at 44, and the subscriber is then notified at 46.

A lower level description of this procedure now follows.

Activation of the create portal button calls the stored procedure spInsertManual to insert a blank record into each of the SQL tblRequest... tables and returns the lRequestID, which is unique to that specific portal. The

portal template information and data that is entered by the end user is then exported via update queries to the tblRequest records just created. The IRequestId is used to reference the correct portal creation records in the event that numerous users are creating portals simultaneously.

When the portal template information and data is written to the tblRequest tables, the procedure spCreateRecords is called, passing the current IRequestId as an identifier for the portal being created. This process commences at the top of the tblRequestPortal for each IRequestId, if a specific IRequestId is not passed.

The spCreateRecords process inspects details of the 'Create Portal' request, (bAddMorePages, IWebsiteID, iRequestQty) in the underlying tblRequest... tables and creates the table records via three sub procedures. These are spCreatePortal, spCreatePortalChild, and spCreateOther. The spCreatePortal procedure creates the portal 'main page' records and also calls spCreateWebsiteChild and spCreateOther. This procedure accepts the IRequestId and iChildQty parameters so that it can process the correct request record and flag each Website as individual web site or parent web site.

The spCreateWebsiteChild procedure solely creates Website records that are flagged as children and attaches them to a parent. This procedure accepts the IRequestId and IWebsiteID parameters so that it can process the correct request record and attach it to the correct Website.

The spCreateOther procedure will create all other Website related records (Announcement, Menu, MenuDeal, ProductService, CataloguePage, and CatalogueItem) and attach them to a Website record. This procedure accepts the IRequestId and IWebsiteID parameters so that it can process the correct request record and attach it to the correct Website.

The password is generated as a string of random letters and digits in a new stored procedure called spCreatePassword. This procedure is called within the spCreatePortal, and spCreateWebsiteChild procedures. This process loops to create the next portals' web site, until the requested number of web sites are created within the portal. After the records are successfully created, the request records that match the passed IRequestId are then deleted from the tblRequest tables.

A scheduled job exists on the Portal Server that will periodically scan the tblRequestWebsiteParent table for records that are flagged as WWW requests (sRequestSource = 'W') and that have the transaction approved (bTransApproved = True).

This routine will loop through all the pending transactions that match the criteria and call the Record Creation procedure for each request. If the Record Creation procedure returns with a Success status, then the user is informed that the request has been processed and will email them with their UserID and Password.

It is expected that during the processing of a portal's creation, this polling routine may be processing hundreds of records at a time.

When records have been created via the WWW, a record is to be written to the tblFaxEmail table so that either a Fax or Email can be sent to the user to inform them that their portal has been created.

The procedures spCreateRecords and spCreatePortal call a procedure called spiEmail, which writes a record to the tblFaxEmail table. An external process then interrogates this table and sends the faxes/emails.

The fax/email process 44 retrieves all 'PENDING' items in the tblFaxEmail table.

The process opens the result set and inspects the sCorrespondanceType field. If it is 'FAX', then a fax file is and sent to the fax server. If it is 'EMAIL', then an email is sent via the ASPMail. If ASPMail detects an error in sending the email, the error message is written to the sFailReason field in the tblFaxEmail table.

Referring now to Figures 3A and 3B, a data entry page is called up from the browser in an HTML format. Alternatively, a locally deployable version of the software, such as Microsoft Visual Basic ®, could be utilized. The number of web sites (in this case ten) is entered together with the affiliate or portal code (CHAMBER), and the business category (Automotive). The number of pages (five) in respect of each web site is also indicated. An order seed no. may also be entered to determine the retrieval priority of the web site. Typically, the web sites are initially allocated the same seed no. so that they have the same priority. Thereafter, individual subscribers may pay a premium for a lower number, so that their web sites have priority when the portal is searched. A logo path (/chamber/images/logo.jpg) is generated for the entire portal based on the affiliate or portal code. As individual web site images are created, they are allocated separate directories within the path. Various other contact details are then entered where applicable, in place of default and prompt data.

In Figure 4, a typical screen layout of a content page 20.2 is shown, which includes the total number of desired content pages (five) as well as an order seed number. Provision is made for the entry of product and service details, together with a product or service external URL.

Once all the other necessary data has been entered by the user in respect of the other pages, the web site user name and password are generated in the manner described above.

At this stage the umbrella organisation has ten web sites carrying separate id's and passwords. This organisation has an administration password enabling it to access and update any web site in the portal. All of the members are then notified of their passwords and ids to enable them to update their individual web sites and enter user specific information.

The updating processes are illustrated in more detail in Figures 6 and 7.

At the customer terminal 16.1, the customer logs into the "Chamber" portal web site via the web browser 17.1 by entering his or her password which is allocated by the umbrella organisation. Together with the updating of business details, the user may update content pages, catalogue pages, announce pages, web site details and cuisine pages. From Figure 7, the various data entry fields are self-explanatory. Once the user has updated the data, the save button is pressed, thereby passing the fields as parameters to the update procedures specific to each page.

As is clear from Figure 6, the update parameters are used to update the corresponding tables in the portal database 12 using portal update procedures 50. The current database information stored in the tables is then transmitted back to the user terminal 16.1 via ASPs, which determine the identical web page layouts. The pages

are then physically viewed by the user at 52. A typical main web page incorporating the information of Figure 6 is shown in Figure 7, with one-to-one correspondence between fields within the database and displayed information units displayed in displaying regions on the web page.

In Figure 8, the searching process is shown. The searches were conducted via a web browser 17.2 at a user terminal 16.2. The search parameters are transmitted via ASPs to a portal search engine 54 at the portal web site. Searching is facilitated within a particular portal using the portal code as part of the search string. The portal code is preserved throughout the user session. The search string is constructed dynamically using the parameters specified during the search. For performance reasons, search criteria not specified are not incorporated within the dynamically created search string within the portal engine. The searcher also has the opportunity to confine the search to a particular web page-defined category within a particular portal. By way of example, business, catalogue, announcement and cuisine searches 54.1 to 54.4 may be conducted, with the search results being conveyed via the portal result pages 56. An example of a search result within a particular portal is illustrated in the page layout of Figure 9.

Referring now to Figure 10, a detailed diagram of the architecture of the portal database is shown. Each of the blocks in the diagram represent individual table location in the database, with the identity of each table being indicated on the diagram. Each of the tables has at least one corresponding web page or part thereof associated with it, with the tables and the web pages having a one-to-one, many-to-one or one-to-many relationship. Similarly, the individual fields within each table house stored information units which have corresponding displayed information units or web regions where the stored information is ultimately displayed. It is clear from the diagram that each of the tables has a particular table identification which is also mirrored in the corresponding web page. The relationship between the tables is indicated by way of the broken lines in the diagram, with a black dot representing the "many" side of a one-to-many relationship.

When a search is conducted in respect of a particular portal, the map field table 58 in the database in respect of that particular portal not only determines the search results, but also the particular fields that are displayed within those results. This allows portals to be customised to the extent that only the relevant data being searched for with respect to a particular portal category is uncovered. Irrelevant information with respect to a particular portal is therefore simply not displayed in the results. The map field table thus determines exactly which information is displayed with respect to a particular portal.

The portal code which is resident in the portal table is used as a primary distinguishing mechanism to distinguish one portal from another within the database. The portal code is preserved for the duration of each user session so as to ensure that only information associated with that particular portal is displayed at the front end.

Various aspects of the present invention are described in more detail, and way of example only, in the following appendices:

Appendix A: Search Engine Specification

Appendix B: Update Admin Specification

Appendix C: Portal Creation Process Definition

Appendix D: Search Architecture ASP Pages

Appendix E: Portal Creation Processes & Functions

5 Although the invention has been described with reference to specific examples, it will be appreciated by those skilled in the art that the invention may be embodied in many other forms.

It will be understood that the invention disclosed and defined herein extends to all alternative combinations of two or more of the individual features mentioned or evident from the text or drawings. All of these different combinations constitute various alternative aspects of the invention.

10 The foregoing describes embodiments of the present invention and modifications, obvious to those skilled in the art can be made thereto, without departing from the scope of the present invention.

Claims

1. A web page including displayed information wherein at least some of the displayed information corresponds with stored information which is stored in fields in tables in a web page database, wherein the displayed information includes a plurality of displayed information units. the stored information includes a plurality of stored information units. and each displayed information unit is displayed in a displaying region on the web page, and wherein the web page database includes a primary web page table including a primary web page field which contains a primary web page database identifier which uniquely identifies the web page database.

2. A web page as claimed in claim 1 wherein each displaying region corresponds to a field in the web page database such that the displayed information unit (DIU) displayed in that region corresponds with the stored information unit (SIU) stored in its corresponding field in the database.

3. A web page as claimed in either one of claims 1 or 2 wherein the position, size, colour and/or orientation of the display regions on the page are defined by a page layout.

4. A web page as claimed in any one of the preceding claims wherein the fields store SIUs chosen from a group including:

- a) characters;
- b) numbers;
- c) pictures;
- d) links; and
- e) dates.

5. A web page as claimed in claim 4 wherein the SIUs are chosen from a group including:

- a) names;
- b) addresses;
- c) phone numbers;
- d) fax numbers;
- e) email addresses;
- f) industry categories;
- g) product names;
- h) product descriptions;
- i) service names;
- j) service descriptions;
- k) announcements;
- l) catalogues;
- m) menus;
- n) keywords;
- o) event names;
- p) event descriptions;

- q) event dates;
- r) pictures; and
- s) banner advertisements.

6. A web page as claimed in any one of the preceding claims wherein the display regions display

5 DIUs are chosen from the group including:

- a) characters;
- b) numbers;
- c) pictures; and
- d) links.

10 7. A web page as claimed in claim 6 wherein the display regions are chosen from a group including:

- a) name regions;
- b) address regions;
- c) phone number regions;
- d) fax number regions;
- 15 e) email address regions;
- f) industry category regions;
- g) product name regions;
- h) product description regions;
- i) service name regions;
- 20 j) service description regions;
- k) announcement regions;
- l) catalogue regions;
- m) menu regions;
- n) keyword regions;
- 25 o) event name regions;
- p) event description regions; or
- q) event date regions;
- r) owner contact information regions;
- s) picture regions;
- 30 t) product information regions;
- u) announcement information regions;
- v) link information regions;
- w) menu information regions; and
- x) banner advertisement regions.

35 8. A web page as claimed in any one of the preceding claims wherein all of the displayed information corresponds with stored information which is stored in the fields in the tables in the web page database.

9. A web page as claimed in any one of the preceding claims wherein the primary web page table is linked to at least one subsidiary web page table by a foreign key.

10. A web page as claimed in claim 9 wherein the at least one subsidiary web page table is linked to further subsidiary tables by further foreign keys.

5 11. A web site including a plurality of web pages as claimed in any one of claims 1 to 10 wherein each web page references, or is referenced by, at least one other web page via a link.

12. A web site including a plurality of web pages, wherein each web page includes displayed information where at least some of the displayed information corresponds with stored information which is stored in fields in tables in a web page database, wherein the displayed information includes a plurality of displayed information units, the stored information includes a plurality of stored information units, and each displayed information unit is displayed in a displaying region on the web page, and, wherein the web site has a unique web site identifier which corresponds to a unique primary web site database identifier such that the displayed information on a particular web site is stored in its corresponding web site database, and wherein each web page references, or is referenced by, at least one other web page via a link.

15 13. A web site as claimed in either one of claims 11 or 12 wherein the displaying regions on the web pages are arranged according to a plurality of page layouts.

14. A web site as claimed in claim 13 wherein the types of page layouts include:

- 20 a) home page layouts;
b) menu page layouts;
c) newsletter page layouts;
d) announcement page layouts;
e) product page layouts;
f) service page layouts;
g) link page layouts;
25 h) general information page layouts; and
I) content page layouts.

15. A web site as claimed in any one of claims 11 to 14 wherein the web site has at least one relational identifier, with the relational identifier of a particular web site being stored in its corresponding web site database.

30 16. A web site as claimed in claim 15 wherein the at least one relational identifier is chosen from the group including:

- 35 a) an affiliate identifier;
b) a parent identifier;
c) a child identifier;
d) a vertical portal identifier; and

- e) a horizontal portal identifier.

17. A set of **affiliate web sites** including a plurality of web sites as defined in claim 16 wherein each web site has the same affiliate identifier.

18. An **hierarchical set of affiliate web sites** including a plurality of affiliate web sites as claimed in claim 17 wherein a parent web site includes at least one child identifier and a child web site includes a parent identifier, such that the child identifier of the parent web site corresponds to the parent identifier of the child web site.

19. A **computer implemented method of forming the web page** as claimed in any one of claims 1 to 10 by:

- a) defining a database including at least one table including at least one field; and
b) associating at least one of the fields with at least one display region on the web page.

20. A **computer implemented method of forming a plurality of web pages** as claimed in any one of claims 1 to 10 by:

- a) creating a template web page including a plurality of display regions whose size, orientation and/or layout is defined by a template page layout;
b) creating a template database including at least one table including a plurality of fields containing default field information and being associated with one or more display regions on the template web page;
c) copying the template database to create a new database having substantially the same table attributes, field attributes and default field information as the template database;
d) creating a new primary database identifier and storing it in a primary field of a primary table of the new database;
e) copying the template web page to create a new web page having a unique web site identifier which corresponds with the new primary database identifier and having a plurality of display regions whose size, orientation and/or layout matches the template page layout;
f) associating the fields of the new database with the display regions on the new web page; and
g) repeating steps b) to d) until the desired number of new web pages is reached.

21. A method as claimed in claim 20 wherein the step of creating a template web page includes the step of accepting layout specifications from a user and applying those specifications to the web page.

22. A **computer implemented method of forming a web site** as claimed in any one of claims 11 to 16 by:

a) forming a first web page in accordance with the method defined in either one of claims 19 or 20;

b) forming a second web page in accordance with the method defined in either one of claims 19 or 20; and

5 c) adding a link, which references the second web page to the first web page.

23. A method as claimed in claim 22 further including the additional step of adding a link, which references the first web page to the second web page .

24. **A computer implemented method of creating and adding a new web page to an existing web site**, wherein the existing web site is a web site as defined in any one of claims 11 to 18, by:

10 a) forming a new web page in accordance with the method defined in any one of claims 19 or 20;

b) adding a first new link, which references the new web page to a first existing web page in the existing web site .

15 25. A method as claimed in claim 24 further including the additional step of adding a second new link, which references the new web page, to a second existing web page.

26. A method as claimed in either one of claims 24 or 25 further including the additional step of adding a third new link, which references an existing web page to the new web page.

27. **A computer implemented method of forming a set of affiliate web sites** as defined in claim 17 by:

20 a) forming a first web site using the method defined in either one of claims 22 or 23;

b) forming a second web site using the method defined in either one of claims 22 or 23; and

c) assigning the same affiliate identifier to the first and second web sites.

28. **A computer implemented method of forming an hierarchical set of affiliate web sites** as defined in claim 18 by:

25 a) forming a set of affiliate web sites using the method defined in claim 27;

b) assigning a child identifier to the parent web site; and

c) assigning a matching parent identifier to the child web site.

29. **A data entry interface for updating displayed information on a web page** as claimed in any one of claims 1 to 10 including a plurality of information entry fields, each of which corresponds to a field in a table in the database.

30 30. An interface as claimed in claim 29 further including an uploader for uploading entered information from the information entry fields into the stored information fields in the database.

31. An interface as claimed in claim 29 further including a downloader for downloading stored information from the stored information fields in the database into the information entry fields.

32. An interface as claimed in claim 30 further including a downloader for downloading stored information from the stored information fields in the database into the information entry fields.

5 33. An interface as claimed in any one of claims 29 to 32 wherein the entered information includes a plurality of entered information units (EIUs).

34. An interface as claimed any one of claims 29 to 33 wherein the plurality of information entry fields are displayed on a plurality of information entry pages.

35. An interface as claimed in claim 34 wherein the information entry pages are internet web pages.

10 36. An interface as claimed in either one of claims 34 or 35 wherein at least some of the information entry pages are linked to one another.

37. An interface as claimed in any one of claims 34 to 36 wherein the information entry pages are chosen from a group including:

- 15 a) personal information entry pages;
b) product information entry pages;
c) service information entry pages;
d) announcement information entry pages;
e) catalogue information entry pages;
f) menu information entry pages;
20 g) general information entry pages;
h) additional links information entry pages; and/or
i) homepage information entry pages.

38. A interface of claim 37 wherein the information entry pages further include action choice buttons.

25 39. The interface as claimed in claim 38 wherein the action choice buttons are links or pictures which behave like buttons.

40. The interface of claim 38 wherein the action choice buttons are chosen from a group including:

- a) edit buttons;
b) add buttons;
c) save buttons; and
30 d) activate pick lists.

41. The interface as claimed in claim 40 wherein pressing the edit button on an information entry page causes the downloader to download the SIUs from the database fields into the corresponding information entry fields in the information page for editing by the user.

42. The interface as claimed in claim 40 wherein pressing the add button on an information entry page causes the downloader to download the SIUs from the database fields into the corresponding information entry fields in the information page in read only format, so that the user can only add entered information to empty information entry fields.

43. The interface as claimed in either one of claims 41 or 42 wherein pressing the save button on an information entry page causes the uploader to upload the entered information units (EIUs) entered by user into the corresponding database fields as SIUs.

44. The interface as claimed in any one of claims 41 to 43 wherein selecting the activate option in the activate pick list on an information entry page causes the uploader to change an active indicator in a field in the database to an active state.

45. **A computer implemented method of generating a search string** for use in searching a database having a plurality of database tables each of which has a database table name, wherein each table contains a plurality of database fields each of which has a database field name and contains stored information, wherein said search string is based upon at least one search parameter, each search parameter having a corresponding database field and wherein said method includes the steps of:

- a) generating a database field selection statement;
- b) generating a database table selection statement;
- c) generating a condition statement;
- d) concatenating said database field selection statement, said database table selection statement and said condition statement to form a search string.

46. A method as claimed in claim 45 further including the additional step of generating a result order statement and concatenating said result order statement to said search string.

47. A method as claimed in claim 45 wherein said step of generating a database field selection statement is achieved by:

- a) declaring a database field selection string variable;
- b) initialising said string variable;
- c) adding a database field selection command to the start of said string;
- d) adding at least one database field name to said string, followed by a separator;
- e) repeating step d) until the field selection string variable includes all the required database field names; and
- f) removing the final separator in the database field selection string.

48. A method as claimed in claim 47 wherein said database field selection command is the SQL selection command "SELECT".

49. A method as claimed in claim 47 wherein said separator is a comma.

50. A method as claimed in claim 45 wherein said step of generating a database table selection statement is achieved by:

- a) declaring a database table selection string variable;
- b) initialising said string variable;
- c) adding a database table selection command to the start of said string;
- d) adding at least one database table name to said string, followed by a separator;
- e) repeating step d) until the table selection string variable includes all the required database table names; and
- f) removing the final separator in said database table selection string.

51. A method as claimed in claim 50 wherein said database field selection command is the SQL selection command "FROM".

52. A method as claimed in claim 50 wherein said separator is a comma.

53. A method as claimed in claim 45 wherein said step of generating a condition statement is achieved by:

- a) declaring a condition string variable;
- b) initialising said string variable;
- c) adding a condition command to the start of said string;
- d) adding at least equation to said string, followed by a separator;
- e) repeating step d) until the condition string variable includes all the required equations;
- and
- f) removing the final separator from the string.

54. A method as claimed in claim 53 wherein said condition command is the SQL selection command "WHERE".

55. A method as claimed in claim 53 wherein said equation takes the form of:

"database field name = corresponding search parameter".

56. A method as claimed in claim 53 wherein said separator is the SQL condition separator "AND".

57. A method as claimed in claim 46 wherein said step of generating a result order statement is achieved by:

- a) declaring a result order string variable;
- b) initialising said string variable;
- c) adding a result order command to the start of said string;
- d) adding at least one database field name to said string, followed by a separator;
- e) repeating step d) until the result order string variable includes all the required database field names; and
- f) removing the final separator from the string.

58. A method as claimed in claim 57 wherein said result order command is the SQL result order command "ORDER BY ".

59. A method as claimed in claim 57 wherein said separator is a comma.

60. A **search string** generated by the method defined in any one of claims 45 to 59.

5 61. A computer implemented **search string generator** which generates search strings in accordance with the method defined in any one of claims 45 to 59.

62. A **web page database** including stored information wherein at least some of the stored information corresponds with displayed information which is displayed in displaying regions on a web page, wherein the web page is a web page as defined in any one of claims 1 to 10.

10 63. A **web site database** including a plurality of web page databases as claimed in claim 62.

64. A web site database including a plurality of web sites as claimed in any one of claims 11 to 18.

65. A web site database as claimed in claim 63 wherein at least some of the stored information in the web site database corresponds with displayed information displayed on the web site as claimed in any one of claims 11 to 16.

15 66. A **super database** including a plurality of web site databases as claimed in claim 64 or claim 65.

67. A super database as claimed in claim 66 wherein each of said web site databases includes an affiliate field which contains an affiliate identifier.

68. A super database as claimed in claim 67 wherein web sites which are affiliated with one another have the same affiliate identifier.

20 69. A super database as claimed in any one of claims 66 to 68 wherein each web site database further includes a visibility field containing a visibility variable which has a positive status and a negative status, indicating whether or not the web site database is visible in the super database.

70. A super database as claimed in any one of claims 66 to 69 wherein said super database includes a super affiliate field which contains a super affiliate code.

25 71. An **internet searching system** including:

a search interface including at least one search field for receiving user supplied search parameters;

a super database as claimed in any one of claims 66 to 70;

a search engine;

30 a search results displaying interface for displaying a plurality of search result units;

wherein said search interface receives said user supplied search parameters and communicates them to said search engine;

said search engine receives said search parameters, generates a query based thereon, searches said super database using said query, produces said plurality of search result units and communicates same to said search results displaying interface; and

wherein said search results displaying interface receives and displays said search result units.

5 72. An internet searching system as claimed in claim 71 wherein said search interface has a plurality of search interface types.

73. An internet searching system as claimed in claim 72 wherein each of said search interface types has an associated search interface identifier.

10 74. An internet searching system as claimed in claim 73 wherein said search interface communicates its search interface identifier to said search engine along with said user supplied search parameters.

75. An internet searching system as claimed in claim 74 wherein said search engine generates an amended query based upon both said search parameters and said search interface identifier.

76. An internet searching system as claimed in claim 75 wherein said search engine searches said super database using said amended query, producing a plurality of filtered search result units.

15 77. An internet searching system as claimed in claim 76 wherein said filtered search result units include stored information which is relevant to said search interface type.

78. An internet searching system as claimed in any one of claims 72 to 77 wherein said search interface type is a super search interface having a super search interface identifier.

20 79. An internet searching system as claimed in claim 78 wherein said amended query includes a condition that the only web site databases which are searched are those having visibility variables with a positive status.

80. An internet searching system as claimed in any one of claims 72 to 77 wherein said search interface is an affiliate search interface having an affiliate search interface identifier.

25 81. An internet searching system as claimed in claim 80 wherein said amended query includes a condition that the only web site databases which are searched are those having an affiliate identifier which corresponds with said affiliate search interface identifier.

82. An internet searching system as claimed in any one of claims 71 to 81 wherein said search results displaying interface has a plurality of interface types.

30 83. An internet searching system as claimed in claim 82 wherein each of said displaying interface types has an associated displaying interface layout.

84. An internet searching system as claimed in claim 83 wherein said displaying interface layout determines the presentation, colour and arrangement of said search result units on said displaying interface.

85. An internet searching system as claimed in claim 84 wherein said search result units are displayed in a search results table having column headings on said displaying interface and wherein said displaying interface layout determines the presentation, colour and arrangement of said results table.

86. An internet searching system as claimed in claim 85 wherein said displaying interface layout
5 determines the information displayed in the column headings of said results table.

87. An internet searching system as claimed in any one of claims 84 to 86 wherein said displaying interface layout further determines the colour, size, orientation and appearance of logos and advertising banners on said displaying interface.

88. An internet searching system as claimed in any one of claims 83 to 87 wherein said displaying
10 interface type is determined by the search interface type used.

89. An internet searching system as claimed in claim 88 wherein said search interface type is a super search interface and said displaying interface type is a super search results displaying interface.

90. An internet searching system as claimed in claim 88 wherein said search interface type is an affiliate search interface and said displaying interface type is an affiliate search results displaying interface.

91. An internet searching system as claimed in any one of claims 71 to 90 wherein said query is a
15 search string as claimed in claim 110.

92. An internet searching system as claimed in claim 82 wherein said search results displaying interface includes at least one search field to enable a user to refine their search.

93. A computer program product comprising a computer readable medium having thereon computer
20 program code means, when said program is loaded, to make the computer execute procedures to provide a data entry interface as claimed in any one of claims 29 to 44.

94. A server hosting a web page database as claimed in claim 62.

95. A server hosting a web site database as claimed in any one of claims 63 to 65.

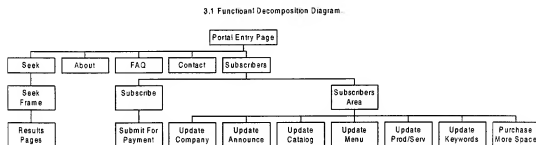
96. A server hosting a super database as claimed in any of claims 66 to 70.

Appendix A Netseek

Netseek Search Engine Functionality Module Functional Specification

1 FUNCTIONAL DECOMPOSITION

1.1 FUNCTIONAL DECOMPOSITION DIAGRAM



1.2 FUNCTION DECOMPOSITION DESCRIPTION

No.	Function	Priority	Description
1	Portal Entry Page	High	Home Page. This page is the main entry page for a Netseek portal. The PortalCode will be hardcoded into this page and will be preserved through the users session. Every system call will require a portal code to be passed from the front end.
2	Seek Frame	High	This page determines how the searches are to be conducted and performs the actual search. This page always remembers the portal code.
	Results Pages	High	This page does display the retrieved results, and allows the user to go to the following pages: company, products/services/content, announcement, catalogue, menu, deal and fax/email pages.
3	Contact	High	This page does display contact details.
4	About Netseek	High	This page does display information about Netseek.
5	FAQ	High	Static Page of FAQ. Help hot link is preferably displayed here too.
6	Subscribers	High	Subscribers area
7	Subscribe To Netseek	High	This page does allow businesses to subscribe to Netseek, and make their payment via AMEX, VISA, MasterCard, and Bankcard. They then Login to the Update area to update their details. The system does issue them with a user name and password.
8	Submit For Payment	High	This page does confirm the payment details and submits the payment to Netseek.
9	Subscribers Area	High	This page does prompt the user for a username and password. On successful validation, the user is preferably presented with the update form.
10	Update Company	High	Allows the user to update their company details.
11	Update Products /	High	Allows the user to add and update products and

No.	Function	Priority	Description
	Services		services. A maximum of 10 products and services is preferably allowed.
12	Update Keywords	High	Allows the user to add and update keywords.
13.	Update Announcement	High	Allows the user to add new announcements or update non-published announcements.
14.	Update Catalogue	High	Allows the user to add new catalogs or update non-published catalogs.
15.	Update Menu	High	For restaurants, allows the user to add or update menu items to their menu.
16.	Purchase Additional Publication Space	High	Allows the user to purchase additional Announcements, Catalogs and Menus.

2 VISUAL DESIGN

The following are visual designs to reflect preferred functionality and purpose.

2.1 INTERFACE STANDARDS

The following are the common elements that appear as standard functions through out the system. It is assumed that the users minimum resolution is preferably 800x600, with 256 colours. It is assumed that either MS Internet Explorer 4.0, Netscape 4.5 or greater is preferably used.

2.2 PAGES

The following are the pages as they appear and descriptions of how they function.

2.2.1 HOME PAGE

This is the main Netseek home page.

No.	Name	Type	Description
1	Seek	Button	This button does activate the search frame, clears the page and displays searching instructions. See 4.2.2 Seek Frame
2	Contact	Button	This button displays Netseek contact details.
3	About Netseek	Button	This button displays the Netseek About page.
4	FAQ	Button	This button displays the Netseek FAQ page.
5	Subscribers	Button	This button displays the Netseek subscribers page.

2.2.2 SEEK FRAME

A search frame appears on the left of each page. This frame contains the following search selection criteria.

No.	Name	Type	Description
1	Category	List Box	The default option is preferably 'All Categories'. The user may select multiple categories.

			If user selects multiple categories, as well as 'All Categories' the system assumes an all category search.
2	Product / Service	List Box	<p>The default option is preferably 'All Products/Services'.</p> <p>The user may select multiple product or services.</p> <p>If user selects multiple product and services, as well as 'All Products/Services' the system assumes an all product/service search.</p>
3	Company	Text Box	<p>The user may enter a company name or part of. Wild cards of % are preferably allowed, such as:</p> <p>'Netse%', or '%etseek', or '%etsee%'.</p> <p>A single % search is not allowed.</p>
4	Seek	Pick List	<p>The user selects one of the following items:</p> <ul style="list-style-type: none"> • Company or Content • Restaurant or Café • Announcement • Catalogue <p>This selection drives the database results returned to the result form.</p>
5	Announcement Type	Pick List	<p>The default option is preferably 'All Announcements'.</p> <p>The user may select a type of announcement, such as 'Sale' or, 'Press Release', etc.</p> <p>This item is preferably only used in a search where Seek = 'Announcement'</p>
6	Catalogue Type	Pick List	<p>The default option is preferably 'All Catalogues'.</p> <p>The user may select a type of catalogue, such as 'Release Estate' or, 'Car', etc.</p> <p>This item is preferably only used in a search where Seek = 'Catalogue'</p>
7	Cuisine Type	Pick List	<p>The default option is preferably 'All Cuisines'</p> <p>The user may select a type of menu, such as 'Thai', or 'Italian', or 'Chinese', etc.</p> <p>This item is preferably only used in a search where Seek = 'Menu'</p>
8	State	Pick List	<p>The Default option is preferably 'NSW'.</p> <p>This pick list is preferably hard-coded. The list is preferably as follows:</p> <p>All States NSW VIC</p>

			<p>ACT SA WA TAS NT</p> <p>The default is preferably 'NSW'</p> <p>The user may select one of these items.</p>
9	Suburb/Postcode	Text Box	<p>The user may enter a suburb or a postcode. If a valid suburb or postcode is not entered, the system assumes all suburbs.</p> <p>Numerous postcodes or suburbs are preferably allowed, a comma can separate them.</p>
10	Keywords	Text Box	<p>The user may enter keywords, which the company may be associated with.</p>
11	Results Returned	Pick List	<p>The default option is preferably 10. The following items are preferably available for selection.</p> <ul style="list-style-type: none"> • 10 • 25 • 50 • 100 <p>This selection does determine the number of records returned to the result page.</p>
13	Seek Buttons	Image / Button	<p>Pressing each SEEK button calls the SP that constructs the search string and returns the results matching the selected criteria. This also parses the PortalCode to the SP.</p>

2.2.3 BASIC SEEK FUNCTIONALITY

No.	Selection	Results
1	Category	<p>All web site records which match the selected category, and status = 'ACTIVE' for the specified portal code.</p> <p>ASCENDING Order By:</p> <p>OrderSeed + CompanyName</p>
2.	Category Product/Service/Information	<p>All web site records which belong to the entered category, and provide the selected products and services, and status = 'ACTIVE' for the specified portal code.</p> <p>ASCENDING Order By:</p> <p>OrderSeed + CompanyName</p>
2a	Company/Web Site	<p>All web site records that match the Company criteria entered, and status = 'ACTIVE' for the</p>

		specified portal code. ASCENDING Order By: Order Seed + Company Name
2b	Keywords	All company records that have associated keywords, and status = 'ACTIVE' for the specified portal code.
3	Seek = Announcement	All announcements for the specified portal code where: Announce End Date<=The System Date, and status = 'ACTIVE' DESCENDING Order By: Announce Start Date ASCENDING Order By: Announce Category Name Company Name
4	Seek = Catalogue	All catalogues for the specified portal code Where: Catalogue End Date<=The System Date, and status = 'ACTIVE' DESCENDING Ordered By: Catalogue Start Date ASCENDING Order By: Catalogue Category Name Company Name
5	Seek = Menu	All restaurants or cafes for the specified portal code where: Menu End Date<=The System Date, and status = 'ACTIVE' ASCENDING Ordered By: Menu Style Name Company Name Menu Page Name

2.2.4 DETAILED SEEK FUNCTIONALITY

No.	Selection	Results
1	Category Seek = Announcement	As outlined in 4.2.3.3, for parent records that are associated with the selected Category for the specified portal code.
2	Category Product/Service/Information Seek = Announcement	As outlined in 4.2.4.1, for parent records that are associated with the selected products and services for the specified portal code.

3	Category Product/Service/Information Seek = Announcement Company Name	As outlined in 4.2.4.2, for parent records that match the name criteria entered for the specified portal code.
4	Category Product/Service/Information Seek = Announcement Company Name State	As outlined in 4.2.4.3, for parent records that match the state selected for the specified portal code.
5	Category Product/Service/Information Seek = Announcement Company Name State Suburb/Postcode	As outlined in 4.2.4.4, for parent records that match the postcode or suburb entered for the specified portal code.
6	Category Seek = Catalogue	As outlined in 4.2.3.4, for parent records that are associated with the selected Category for the specified portal code.
7	Category Product/Service/Information Seek = Catalogue	As outlined in 4.2.4.6, for parent records that are associated with the selected products and services for the specified portal code.
8	Category Product/Service/Information Seek = Catalogue Company Name	As outlined in 4.2.4.7, for parent records that match the name criteria entered for the specified portal code.
9	Category Product/Service/Information Seek = Catalogue Company Name State	As outlined in 4.2.4.8, for parent records that match the state selected for the specified portal code.
10	Category Product/Service/Information Seek = Catalogue Company Name State Suburb/Postcode	As outlined in 4.2.4.9, for parent records that match the postcode or suburb entered for the specified portal code.
11	Category Seek = Menu	As outlined in 4.2.3.5, for parent records that are associated with the selected Category for the specified portal code.
12	Category Product/Service/Information Seek = Menu	As outlined in 4.2.4.11, for parent records that are associated with the selected products and services for the specified portal code.
13	Category Product/Service/Information Seek = Menu Company Name	As outlined in 4.2.4.12, for parent records that match the name criteria entered for the specified portal code.
14	Category Product/Service/Information Seek = Menu Company Name State	As outlined in 4.2.4.13, for parent records that match the state selected for the specified portal code.
15	Category Product/Service/Information Seek = Menu Company Name State	As outlined in 4.2.4.14, for parent records that match the postcode or suburb entered for the specified portal code.

	Suburb/Postcode	
--	-----------------	--

2.2.5 SEEK RESULTS GRID PROPERTIES

The following fields are preferably displayed in the result grid generated from a search. These will be driven by a table called tblPortalResultsFields. This table will record the field names and results column names to be specified in any of the following result sets. The software will work in such a way, where prior to returning a set of results, it will determine the fields which are required for view, and present them in the result columns.

No.	Seek Results Requested	Fields Displayed
1	Company/Main Page	Company Name/Page Title Page Category Name Main Phone Contact Title + Contact F Name + Contact S Name Suburb State
2	Announcement	Company Name Announcement Category Name Announcement Title Main Phone Contact Title + Contact F Name + Contact S Name Suburb State
3	Catalogue	Company Name Catalogue Category Name Catalogue Name Main Phone Contact Title + Contact F Name + Contact S Name Suburb State
4.	Restaurant or Cafe	Restaurant Name Cuisine Name Main Phone Contact Title + Contact F Name + Contact S Name Suburb State

2.2.6 SEEK RESULTS PAGE PROPERTIES

The following fields are preferably displayed when the user selects a result record. Please note that fields in *italics* do not have their field labels shown, and are preferably centered on the page.

No.	Results Shown	Fields Displayed
1	Company/Main Page	Company Name Business Category Name Street1 Street2 Suburb Postcode State Main Phone Main Fax Contact Title + Contact F Name + Contact S Name Contact Phone

		<p>Contact Fax Contact Mobile Contact Email <i>Company Description Header</i> <i>Company Description Body</i> <i>Company Description Footer</i> <i>URL</i></p> <p>Links To Be Created On Left Hand Side:</p> <p><Parent Company> (If any exist) List of <Products/Services/Info Pages> (If any exist) List of <Menu Pages> (If any exist) <Announcements> Label (If any exist) <Catalogues> Label (If any exist) <Fax Company></p>
2	Announcement	<p>Company Name Business Category Name Street1 Street2 Suburb Postcode State Main Phone Main Fax Contact Title + Contact F Name + Contact S Name Contact Email URL Announce Category Name <i>Announce Title</i> <i>Announce Sub-title Text</i> <i>Announce Text</i> <i>Announce Page Footer</i></p> <p>Buttons:</p> <p><Company> <Products/Services> <Catalogues> (If any exist) <Menus> (If any exist) <Fax Company></p>
3	Catalogue	<p>Company Name Business Category Name Street1 Street2 Suburb Postcode State Main Phone Main Fax Contact Title + Contact F Name + Contact S Name Contact Email URL <i>Catalogue Name</i> <i>Catalogue Description</i> <i>Catalogue Image 1</i> <i>Catalogue Image 1 Description</i> <i>Catalogue Image 2</i> <i>Catalogue Image 2 Description</i> <i>Catalogue Image 3</i></p>

		<p><i>Catalogue Image 3 Description</i> <i>Catalogue Image 4</i> <i>Catalogue Image 4 Description</i> <i>Catalogue Image 5</i> <i>Catalogue Image 5 Description</i> <i>Catalogue Image 6</i> <i>Catalogue Image 6 Description</i> <i>Catalogue Page Footer</i></p> <p>Buttons:</p> <p><Company> <Products/Services> <Announcements> (If any exist) <Menus> (If any exist) <Fax Company></p>
4.	Menu	<p>Company Name Business Category Name Street1 Street2 Suburb Postcode State Main Phone Main Fax Contact Title + Contact F Name + Contact S Name Contact Email URL</p> <p>Menu List Title: Item Description, Small, Medium, Large</p> <p><i>Menu Item 1 Image</i> <i>Menu Item 1 Description</i> <i>Menu Item 1 Small Price</i> <i>Menu Item 1 Medium Price</i> <i>Menu Item 1 Large Price</i></p> <p><i>Menu Item 2 Image</i> <i>Menu Item 2 Description</i> <i>Menu Item 2 Small Price</i> <i>Menu Item 2 Medium Price</i> <i>Menu Item 2 Large Price</i></p> <p><i>Menu Item 3 Image</i> <i>Menu Item 3 Description</i> <i>Menu Item 3 Small Price</i> <i>Menu Item 3 Medium Price</i> <i>Menu Item 3 Large Price</i></p> <p><i>Menu Item 4 Image</i> <i>Menu Item 4 Description</i> <i>Menu Item 4 Small Price</i> <i>Menu Item 4 Medium Price</i> <i>Menu Item 4 Large Price</i></p> <p><i>Menu Item 5 Image</i> <i>Menu Item 5 Description</i> <i>Menu Item 5 Small Price</i> <i>Menu Item 5 Medium Price</i></p>

		<p><i>Menu Item 5 Large Price</i></p> <p><i>Menu Item 6 Image</i> <i>Menu Item 6 Description</i> <i>Menu Item 6 Small Price</i> <i>Menu Item 6 Medium Price</i> <i>Menu Item 6 Large Price</i></p> <p>Buttons:</p> <p><Company> <Products/Services> <Announcements> (If any exist) <Catalogues> (If any exist) <Fax Company></p>
--	--	--

2.2.7 WEB PAGE NAVIGATION BAR.

When a Netseek web site page is displayed. The left hand bar is dynamically constructed based on the results returned by spgmPickLists. This allows every netseek web site to have its own established navigation bar, which is created by various entries in it tables, such that everything is data driven.

2.2.7.1 CONTACT

The Fax Company button does present the user with a fax or email data entry form containing the following fields:

No.	Name	Type	Description
1	To	Text	This field displays the contact name as default, but may be changed by the user.
2	To Phone	Text Read Only	This field does default to the phone number of the company contact.
3	To Fax	Text Read Only	This field does default to the fax number of the company contact.
4	To Email	Text Read Only	This field does default to the email address of the company contact.
5.1	Sender List	Picklist	This list is to be populated with all the contacts that exist within the portal. Sender Name, Sender Phone, and Sender Fax will be automatically populated on selection. Initial value will default to N/A.
5.2	Sender Name	Text	The user does enter their name.
6	Sender Phone	Text	The user does enter their contact phone number.
7	Sender Fax	Text	The user does enter their contact fax number.
8	Date	Date Time Read Only	This does default to the current system date and time.
9	Re	Text	The user does enter what the fax is regarding.
10	Body	Text	The user does type in the body of the fax.
11	Submit	Button	The message is submitted for processing. It will be emailed or faxed depending on how the company is configured.
12	15 Minute Process	Process	A server side process is run every 15 mins automatically to fax or email all messages.

The message is preferably written to a fax table, which records the CompanyID against the record. A process sitting on Fax/Email Server does periodically poll this table. If any new unsent messages have been inserted into this table, they are either emailed or faxed, and then updated as being sent on the server. If an error occurs, the error is recorded against the record so that the cause can be determined.

2.2.8 CONTACT

This page displays the Netseek contact details.

2.2.9 ABOUT NETSEEK

This page displays information about Netseek.

Appendix B

Netseek Pty Ltd Update Admin Specification Module Functional Specification

1 FUNCTIONAL DECOMPOSITION

1.1 FUNCTION DECOMPOSITION DESCRIPTION

No.	Function	Type	Description
1	Subscribers Area	Home Page Link	This page will prompt the user for a username and password. On successful validation, the user will be presented with the update interface.
1.1	Select Website To Update	Link	This option will present the user with a list of all the websites that fall under that portal or portal parent. This option is made available only if they are an administrator – determined by the PortalCode table.
2	Business Details	Link	Allow the user to update their company details.
	Home Page	Link	Allow the user to update their home page details.
3	Content Pages	Link	Allow the user to update products and services.
4	Announcements	Link	Allow the user to update non-active announcements, and set active announcements to inactive. Once an announcement becomes active, it cannot be edited, except to allow a status change to INACTIVE.
5	Catalogues	Link	Allow the user to update active catalogs or update non-active catalogs.
6	Menus	Link	For restaurants, allow the user to update active and non-active menu items on their menu.
7	Deals	Link	For restaurants, allow the user to update active and non-active deal items.
8	User Info	Link	This page will allow the users details to be changed. User name will not be editable ever. It will be system assigned.
9	Web Site Statistics	Link	This page will display web site statistics based on the results returned by an SP.
10	Update Links List	Link	Update the Additional Links table.
11	Purchase A Web Site / Additional Pages	Link	This component integrates with 3 rd party e-commerce systems.
12	Statistics	Link	This page displays page view statistics based on a date range. Each page presence within the portal will have a statistic recorded at the time of page activation.

2 VISUAL DESIGN

2.1 INTERFACE STANDARDS

The following are the common elements that appear as standard functions through out the system. It is assumed that the users minimum resolution will be 800x600, with 256 colours. It is assumed that either MS Internet Explorer 4.0, Netscape 4.5 or greater will be used.

All pages will have Save functionality located in the same position. Such as a save button, or a Cancel button, etc.

2.2 PAGES

The following are the pages as they will appear and descriptions of how they function.

2.2.1 SUBSCRIBERS

This page will provide an interface for accepting new subscribers or maintaining existing subscribers' details.

The following button will be displayed on the home page in the form of an image.

No.	Name	Type	Description
1	Subscribers Area	Button	<p>This will prompt the user to enter their User ID and Password. If validated correctly, and the web site record is either 'ACTIVE' or 'INACTIVE' give the user access to the update area.</p> <p><i>The following stored procedure will be called:</i></p> <pre> spvValidateUser @UserID char(15) , @sPassword varchar(10) If successful, the following resultset will be returned: 1 AS bSuccess, sType, sPortalCode, lCompanyID, lPortalCode otherwise, if unsuccessful: 0 AS bSuccess, sReason </pre> <p>Otherwise do not allow the subscriber into the maintenance area. Redirect to a page requesting they contact the portal owner, or they specify an email address to have their password emailed automatically.</p>

2.2.1.1 VALIDATION SUCCESSFUL - SUBSCRIBERS AREA

The subscriber's area will present the user with a bar on the left of the page that is common throughout the whole system. (Use current bar measurements).

When the bar is initially constructed and displayed, the right hand side will contain instructions on the usage of the system.

The header will be the same as the header on the web site home pages. Room for a banner is required.

As a standard, each page's images will be displayed on that page when editing throughout the whole system.

The Portal Name, user name and the currently selected web site being edited is displayed at the top of the page underneath the header in a grid, which resembles:

Portal: EHEALTH	User Name: Jim Limberis – NS82732
You Are Editing: Specialist Hospitals	

The colours for this grid are driven by the PortalTheme table, which will record the colour schemes supported by the portal.

The following buttons/Links will be displayed:

No.	Name	Type	Description
0	Select Website To Update	Button/Link	This option will present the user with a list of all the websites that fall under that Portal or parent. This link will only be active if the user is a Portal administrator.
1.1	Website Setup	Button/Link	This page will allow the user to select colours and a presence theme for their online presence. A facility to upload a logo is also available.
1.2	Company Details	Button/Link	This page will allow the user to update their company details. When clicked, the left hand bar and links will remain. On the right of the left hand bar, the company fields will be displayed for editing.
2	Main Page	Button/Link	This page will allow the user to update their home page wordings. When clicked, the left hand bar and links will remain. On the right of the bar, the home page fields will be displayed for editing.
3	Content	Button/Link	This page will allow the user to update general content (prod/serv) pages. A list of products and services associated to this web site will be displayed on the right hand side of the bar. (Similar to the result's grid presented in the search). When the user selects one of these records, the fields for that record will be displayed below the list for editing.
4	Announcements	Button/Link	This page will allow the user to update Announcement pages, and will work much the same way as the product/service above.
5	Catalogues	Button/Link	This page will allow the user to update Catalogue pages, and will work much the same way as the product/service above.
6	Menus	Button/Link	This page will allow the user to update Menu pages, and will work much the same way as

			the product/service above.
7	Deals	Button/Link	This page will allow the user to update Deal pages, and will work much the same way as the product/service above.
8	Statistics	Button/Link	This page will display web site statistics based on the results returned by an SP.

2.2.1.2 SELECT WEBSITE TO UPDATE

If the user who logged in is a Portal or a parent, then this link must be made visible. On selection, the user will be presented with a grid containing all the web sites that fall under the Portal or parent code that the user has logged in as.

(If they are a parent, the first record in the results set will be their web site, all others will be listed in alphabetical order based on web page title – this is to be controlled from the SQL Server).

The form will show 15 records at a time. With an option to go <Next> and <Prev> just like a typical search result set. (There may be 100's of web sites under a Portal). A search field will also be made available to allow the user to search the list of portal websites, and thus filter down on their selection.

Once a record is selected, by clicking on the first column, the system will place this record into edit mode and update the grid to reflect the selection. (See the 'You Are Editing:' box).

Portal: EHEALTH	User Name: Jim Limberis – NS82732
You Are Editing: Specialist Hospitals	

2.2.1.3 BUSINESS DETAILS

The company details will be presented for editing. The following fields are to be displayed in this order.

I'd like all fields to be displayed in a table containing three columns, as follows:

1. Name – Left aligned
2. Actual Field – Left aligned
3. Help – Left aligned – To contain text regarding the use of the fields.

The physical form length of each field should vary to reflect the number of characters allowed.

All fields and fonts are to default to the Arial Standard font.

Field Name	Source	Form Label Name	Functionality Required
sCompanyName	tblCompany	Business Name	Must be able to accept apostrophes.
sStreet1	tblCompany	Street	
sStreet2	tblCompany		Positioned in the same grid box as Street1. Directly underneath Street1.
sSuburb	tblCompany	Suburb	First letter capital.
sRegion	tblCompany	Region	Pick list of regions as they appear on the home page.
sPostCode	tblCompany	Post Code	4 numerics.
sState	tblCompany	State	Pick list of states as they appear on the home page.

sMainPhone	tblCompany	Main Phone	
sMainFax	tblCompany	Main Fax	
sContactTitle	tblCompany	Title	Pick list containing 'Mr, Ms, Mrs, Miss, Dr, Prof.'
sContactFName	tblCompany	First Name	First letter capital.
sContactSName	tblCompany	Surname	First letter capital.
sContactPhone	tblCompany	Direct Phone	
sContactFax	tblCompany	Direct Fax	
sContactMobile	tblCompany	Mobile	
SContactEmail	tblCompany	Email	
SURL	tblCompany	Current Website URL	Can the existence of this URL be tested on Entry?
BrequiresFax	tblCompany	Send Messages As Faxes?	Pick list: Yes or No.
SNetseekURL	tblCompany	Netseek URL	This will be the name of the directory created for this business, and will be the directory that its default ASP page as well as the images and image sub directories will be located in.

The following stored procedure will be called for retrieving results:

```
spgCompanyDetail
    @lCompanyID int,
    @sPortalCode udtPortalCode
```

If no sPortalCode is supplied, then the resultset will be filtered where bAllowWrite = 1

The resultset that is returned is as follows:

```
lCompanyID,
sStreet1,
sStreet2,
sSuburb,
sRegion,
sPostCode,
sState,
sMainPhone,
sMainFax,
sContactTitle,
sContactFName,
sContactSName,
sContactPhone,
sContactFax,
sContactMobile,
sContactEmail,
bRequiresFax,
sURL,
sNetseekURL,
sCompanySeekName,
bRequiresMenu,
sCompanyLogoImagePath1,
sPortalCode,
bVisibleToNetseek,
sClientLogoPath,
sClientLogoURL
```

The following stored procedure will be called for updating results:

```

spuCompanyDetail
    @iCompanyID int,
    @sStreet1 varchar(50) = '',
    @sStreet2 varchar(50) = '',
    @sSuburb char(30) = '',
    @sRegion char(19) = '',
    @sPostCode char(4) = '',
    @sState char(3) = '',
    @sMainPhone udtPhone = '',
    @sMainFax udtPhone = '',
    @sContactTitle varchar(5) = '',
    @sContactFName varchar(25) = '',
    @sContactSName varchar(35) = '',
    @sContactPhone udtPhone = '',
    @sContactFax udtPhone = '',
    @sContactMobile udtPhone = '',
    @sContactEmail varchar(50) = '',
    @bRequiresFax udtYesNo = 0,
    @sURL varchar(150) = '',
    @sNetseekURL varchar(75) = '',
    @sCompanySeekName char(50) = '',
    @sCompanyLogoImagePath1 varchar(255) = '',
    @sPortalCode udtPortalCode = 'Netseek',
    @bVisibleToNetseek udtYesNo = 0,
    @sClientLogoPath varchar(100) = '',
    @sClientLogoURL varchar(100) = ''

```

These fields will not be displayed (therefore no Form Label Name), but they must either default to something or perform some sort of function.

Field Name	Functionality Required
sCompanySeekName	Will default to the sCompanyName minus the apostrophes. Ideally the front end should do this, as SQL Server does not like dealing with apostrophes.
bRequiresMenu	IF sBusinessCategoryName = 'Restaurant' or 'Café', then this is automatically set to TRUE.
sCompanyLogoImage FilePath1	This is the full path of the picture file to be displayed for the company home page. It should be displayed as an image. When the user selects to change an image, the system should concatenate the appropriate path string including the name of the file, and save it to this field, although the image should be displayed.

2.2.1.4 HOME PAGE

The home page details will be presented for editing. The following fields are to be displayed in this order.

I'd like all fields to be displayed in a table containing three columns, as follows:

1. Name – Left aligned
2. Actual Field – Left aligned
3. Help – Left aligned – To contain text regarding the use of the fields.

The physical form length of each field should vary to reflect the number of characters allowed.

All fields and fonts are to default to the Arial Standard font.

See NetseekPhase2HomePageLayout.Doc file for the form layout.

Field Name	Source	Form Label Name	Functionality Required
sBusinessCategoryName	tblCompany	Business Category	Data Driven. Per the current system Business Category pick list. (Current field is in Business Seek area)
mCompanyDescHeader	tblCompany	Page Heading	
mCompanyDescBody	tblCompany	Page Main Body Text	This should be at least 15 lines long and 50 chars wide. Enable the ENTER button to perform CR rather than NEXT field.
mCompanyDescFooter	tblCompany	Page Footer	
sCompanyKeywords	tblCompany	Keywords	This should be 5 lines long and 50 chars wide.
sStatus	tblCompany	Website Status	This field is a pick list. The options are: INACTIVE ACTIVE

The following stored procedure will be called for retrieving results:

```
spgCompanyHome
    @lCompanyID int,
    @sPortalCode udtPortalCode = Null
```

If no sPortalCode is supplied, then the resultset will be filtered where bAllowWrite = 1

The resultset that is returned is as follows:

```
lCompanyID,
sBusinessCategoryName,
mCompanyDescHeader,
mCompanyDescBody,
mCompanyDescFooter,
sCompanyKeywords,
sStatus,
sPortalCode,
bVisibleToNetseek,
sClientLogoPath,
sClientLogoURL
```

The following stored procedure will be called for updating results:

```
spuCompanyHome
    @lCompanyID int,
    @sBusinessCategoryName char(19) = '',
    @mCompanyDescHeader varchar(500) = '',
    @mCompanyDescBody varchar(4000) = '',
    @mCompanyDescFooter varchar(500) = '',
    @sCompanyKeywords varchar(500) = '',
    @sStatus udtStatus = 'INACTIVE',
    @sPortalCode udtPortalCode = 'Netseek',
    @bVisibleToNetseek udtYesNo = 0,
    @sClientLogoPath varchar(100) = '',
    @sClientLogoURL varchar(100) = ''
```

Unresolved Issues

Issue Name	Description
Image Upload Functionality	<p>We need to be able to upload images from a user's local drives into the appropriate image directories and subdirectories when a user wants to change an image in a page. A separate directory will exist for each customer in the format of:</p> <p style="text-align: center;">\PortalRoot\CustomerDirectory</p> <p>Each website directory will be data driven from the tblCompany table. This will allow each website to have their own directory.</p>
Change Image	<p>We require the ability to change a page image by selecting from a list of images from a pick list.</p> <p>In the event that a person does not have their own images, they will be able to select from a generic list, which will essentially be a list of files in GlobalImages directory. When displaying files, allow for 35 Chars to display the Name option of the file.</p> <p>So, each Netseek Web Site will have two sources for images, the ones located in the clients directories (which they uploaded) and the ones located in the GlobalImages directory.</p>

2.2.1.5 CONTENT PAGES LIST

The content pages for this web site will be presented (in ascending order of tblCompanyProdServ.iPageOrder) for selection via a results grid of some sorts. It should look and behave similar to the current result's grids in the system. When a user selects a Product / Service page record from the grid, its fields will be displayed below the grid for editing.

These fields are to be displayed in the Product / Service Page list:

Field Name	Source	Heading Label	Functionality Required
sProdServName	tblProdServ	Page Name	Ability to click and select. Clicking on this will show that record for editing below.
sStatus	tblProdServ	Page Status	
mProdServDesc	tblProdServ	Description	First 70 Chars of this field to be displayed only.

The following stored procedure will be called for retrieving results:

```
spgCompanyProdServList
    @lCompanyID int,
    @sPortalCode udtPortalCode = Null
```

If no sPortalCode is supplied, then the resultset will be filtered where bAllowWrite = 1

The resultset that is returned is as follows:

```
tblCompanyProdServ.lProdServID,
tblCompanyProdServ.sProdServName,
tblCompanyProdServ.sStatus,
Left(tblCompanyProdServ.mProdServDesc, 70) AS sShortDesc
```


Once a 'prod/serv' page record is selected, I'd like all fields to be displayed in a table containing three columns, as follows:

1. Name – Left aligned
2. Actual Field – Left aligned
3. Help – Left aligned – To contain text regarding the use of the fields.

The physical form length of each field should vary to reflect the number of characters allowed.

All fields and fonts are to default to the Arial Standard font.

Field Name	Source	Form Label Name	Functionality Required
iPageOrder	tblCompanyProdServ	Page Order	Numeric allow 2 digits to be entered. Only digits.
SProdServCategoryName	tblCompanyProdServ	Category	Data driven pick list. Populate per the current product / service pick list. (Current field is in Business Seek area)
sProdServName	tblCompanyProdServ	Page / Link Name	Must Not Exceed 20 Chrs.
mProdServDesc	tblCompanyProdServ	Page Description	This should be at least 15 lines long and 50 chars wide. Enable the ENTER button to perform CR rather than NEXT field.
sProdServPageFooter	tblCompanyProdServ	Page Footer	
SprodServImageFilePath1	tblCompanyProdServ		This is the full path of the picture file to be displayed for the product/service page. It should be displayed as an image. When the user selects to change an image, the system should concatenate the appropriate path string including the name of the file, and save it to this field, although the image should be displayed.
sStatus	tblCompanyProdServ	Page Status	This field is a pick list. The options are: INACTIVE ACTIVE

The following stored procedure will be called for retrieving results:

```
spgCompanyProdServDetail
    @lProdServID int
```

The resultset that is returned is as follows:

```
tblCompanyProdServ.lProdServID,
tblCompanyProdServ.iOrderSeed,
tblCompanyProdServ.sProdServCategoryName,
tblCompanyProdServ.sProdServName,
tblCompanyProdServ.mProdServDesc,
tblCompanyProdServ.sProdServPageFooter,
```

```
tblCompanyProdServ.sProdServImagePath1,
tblCompanyProdServ.sStatus
```

The following stored procedure will be called for updating results:

```
spuCompanyProdServDetail
    @lProdServID int,
    @iOrderSeed udtOrderSeed = 99,
    @sProdServCategoryName char(19) = '',
    @sProdServName varchar(20) = '',
    @mProdServDesc varchar(4000) = '',
    @sProdServPageFooter varchar(255) = '',
    @sProdServImagePath1 varchar(255) = '',
    @sStatus udtStatus = 'INACTIVE'
```

This page needs image upload and image change functionality like the Update Web site Home Page. The currently selected image should always be displayed.

2.2.1.6 UPDATE ANNOUNCEMENT LIST

The announcement pages for this web site will be presented for selection via a results grid of some sorts. It should look and behave similar to the current result's grids in the system. When a user selects an announcement page record from the grid, its fields will be displayed below the grid for editing.

These fields are to be displayed in the Announcements list:

Field Name	Source	Heading Label	Functionality Required
sAnnounceTitle	tblCompanyAnnounce	Title	Ability to click and select. Clicking on this will show that record for editing below.
sAnnounceSubTitle	tblCompanyAnnounce	Sub Title	Ability to click and select. Clicking on this will show that record for editing below.
sAnnounceCategoryName	tblCompanyAnnounce	Type	
sStatus	tblCompanyAnnounce	Page Status	
tAnnounceStartDate	tblCompanyAnnounce	Start Date	
tAnnounceEndDate	tblCompanyAnnounce	End Date	

The following stored procedure will be called for retrieving results:

```
spgAnnouncementList
    @lCompanyID int,
    @sPortalCode udtPortalCode = Null
```

If no sPortalCode is supplied, then the resultset will be filtered where bAllowWrite = 1

The resultset that is returned is as follows:

```
lAnnounceID,
sAnnounceTitle,
sAnnounceSubTitle,
```

sAnnounceCategoryName,
tAnnounceStartDate,
tAnnounceEndDate

Once an announcement page record is selected, I'd like all fields to be displayed in a table containing three columns, as follows:

4. Name – Left aligned
5. Actual Field – Left aligned
6. Help – Left aligned – To contain text regarding the use of the fields.

The physical form length of each field should vary to reflect the number of characters allowed.

All fields and fonts are to default to the Arial Standard font.

Field Name	Source	Form Label Name	Functionality Required
SAnnounceCategoryName	tblCompanyAnnounce	Type	
sAnnounceTitle	tblCompanyAnnounce	Title	
sAnnounceSubTitle	tblCompanyAnnounce	Sub Title	
sAnnounceText	tblCompanyAnnounce	Description	This should be at least 10 lines long and 50 chars wide. Enable the ENTER button to perform CR rather than NEXT field.
sAnnouncePageFooter	tblCompanyAnnounce	Page Footer	
tAnnounceStartDate	tblCompanyAnnounce	Start Date	Y2K compliant date and format.
tAnnounceEndDate	tblCompanyAnnounce	End Date	Y2K compliant date and format.
sStatus	tblCompanyAnnounce	Page Status	This field is a pick list. The options are: INACTIVE ACTIVE
SAnnounceImageFilePath1	tblCompanyAnnounce		This is the full path of the picture file to be displayed for the announcement page. It should be displayed as an image. When the user selects to change an image, the system should concatenate the appropriate path string including the name of the file, and save it to this field, although the image should be displayed.

The following stored procedure will be called for retrieving results:

```
spgAnnouncementDetail
    @lAnnounceID int
```

The resultset that is returned is as follows:

lAnnounceID,

```

sAnnounceCategoryName,
sAnnounceTitle,
sAnnounceSubTitle,
mAnnounceText,
sAnnouncePageFooter,
tAnnounceStartDate,
tAnnounceEndDate,
sStatus,
sAnnounceImageFilePath1

```

The following stored procedure will be called for updating results:

```

spuAnnouncementDetail
    @lAnnounceID int,
    @sAnnounceCategoryName char(19) = '',
    @sAnnounceTitle varchar(50) = '',
    @sAnnounceSubTitle varchar(100) = '',
    @mAnnounceText varchar(4000) = '',
    @sAnnouncePageFooter varchar(255) = '',
    @tAnnounceStartDate udtDateTime = NULL,
    @tAnnounceEndDate udtDateTime = NULL,
    @sStatus udtStatus = 'INACTIVE',
    @sAnnounceImageFilePath1 varchar(255) = ''

```

This page needs image upload and image change functionality like the Update Web site Home Page. The currently selected image should always be displayed.

2.2.1.7 UPDATE MENU LIST

The menu pages for this web site will be presented (in ascending order of tblCompanyMenu.iPageOrder) for selection via a results grid of some sorts. It should look and behave similar to the current result's grids in the system. When a user selects a menu page record from the grid, its fields will be displayed below the grid for editing.

These fields are to be displayed in the Menu list:

Field Name	Source	Heading Label	Functionality Required
sMenuStyle Name	tblCompany Menu	Cuisine	Ability to click and select. Clicking on this will show that record for editing below.
sMenuPageName	tblCompany Menu	Page Title	Ability to click and select. Clicking on this will show that record for editing below.
sMenuPageDesc	tblCompany Menu	Page Description	
sStatus	tblCompany Menu	Page Status	

The following stored procedure will be called for retrieving results:

```

spgCompanyMenuList
    @lCompanyID int,
    @sPortalCode udtPortalCode = Null

```

If no sPortalCode is supplied, then the resultset will be filtered where bAllowWrite = 1

The resultset that is returned is as follows:

```

1.MenuID,
s.MenuStyleName,
s.MenuPageName,
s.MenuPageDesc,
s.Status

```

Once a menu page record is selected, the menu edit page should be constructed to look similar to the existing menus, with the common fields presented in both the same vertical and horizontal order. All menu images are to be displayed as well.

Field Name	Source	Form Label Name	Functionality Required
iPageOrder	tblCompanyMenu	Page Order	Numeric allow 2 digits to be entered. Only digits.
sMenuStyleName	tblCompanyMenu	Cuisine	
sMenuPageName	tblCompanyMenu	Menu Title	
sMenuPageDesc	tblCompanyMenu	Menu Description	
sMenuItem1ImageFilePath	tblCompanyMenu		This is the full path of the picture file to be displayed for the menu item. It should be displayed as an image. When the user selects to change an image, the system should concatenate the appropriate path string including the name of the file, and save it to this field, although the image should be displayed.
sMenuItem1Desc	tblCompanyMenu	Description	
cMenuItem1LargePrice	tblCompanyMenu	Price	In the format of \$999999.99 where the digit 9 represents any valid integer between 0 and 9.
sMenuItem2ImageFilePath	tblCompanyMenu		This is the full path of the picture file to be displayed for the menu item. It should be displayed as an image. When the user selects to change an image, the system should concatenate the appropriate path string including the name of the file, and save it to this field, although the image should be displayed.
sMenuItem2Desc	tblCompanyMenu	Description	
cMenuItem2LargePrice	tblCompanyMenu	Price	In the format of \$999999.99 where the digit 9 represents any valid integer between 0 and 9.
sMenuItem3ImageFilePath	tblCompanyMenu		This is the full path of the picture file to be displayed for the menu item. It should be displayed as an image. When the user selects to change an image, the system should concatenate the appropriate path string including the name of the file, and save it to this field, although the

			image should be displayed.
sMenuItem3Desc	tblCompany Menu	Description	
cMenuItem3LargePrice	tblCompany Menu	Price	In the format of \$999999.99 where the digit 9 represents any valid integer between 0 and 9.
sMenuItem4ImageFilePath	tblCompany Menu		This is the full path of the picture file to be displayed for the menu item. It should be displayed as an image. When the user selects to change an image, the system should concatenate the appropriate path string including the name of the file, and save it to this field, although the image should be displayed.
sMenuItem4Desc	tblCompany Menu	Description	
cMenuItem4LargePrice	tblCompany Menu	Price	In the format of \$999999.99 where the digit 9 represents any valid integer between 0 and 9.
sMenuItem5ImageFilePath	tblCompany Menu		This is the full path of the picture file to be displayed for the menu item. It should be displayed as an image. When the user selects to change an image, the system should concatenate the appropriate path string including the name of the file, and save it to this field, although the image should be displayed.
sMenuItem5Desc	tblCompany Menu	Description	
cMenuItem5LargePrice	tblCompany Menu	Price	In the format of \$999999.99 where the digit 9 represents any valid integer between 0 and 9.
sMenuPageFooter	tblCompany Menu	Menu Page Footer	
sStatus	tblCompany Menu	Page Status	This field is a pick list. The options are: INACTIVE ACTIVE

The following stored procedure will be called for retrieving results:

```
spgCompanyMenuDetail
    @lMenuID int
```

The resultset that is returned is as follows:

```
lMenuID,
iOrderSeed,
sMenuStyleName,
sMenuPageName,
sMenuPageDesc,
sMenuItem1ImageFilePath,
sMenuItem1Desc,
cMenuItem1LargePrice,
sMenuItem2ImageFilePath,
sMenuItem2Desc,
cMenuItem2LargePrice,
sMenuItem3ImageFilePath,
```

```

sMenuItem3Desc,
cMenuItem3LargePrice,
sMenuItem4ImageFilePath,
sMenuItem4Desc,
cMenuItem4LargePrice,
sMenuItem5ImageFilePath,
sMenuItem5Desc,
cMenuItem5LargePrice,
sMenuItem6ImageFilePath,
sMenuItem6Desc,
cMenuItem6LargePrice,
sMenuItemPageFooter,
sStatus

```

The following stored procedure will be called for updating results:

```

spuCompanyMenuDetail
    @lMenuID int,
    @iOrderSeed udtOrderSeed = 99,
    @sMenuStyleName varchar(19) = '',
    @sMenuPageName varchar(20) = '',
    @sMenuPageDesc varchar(255) = '',
    @sMenuItem1ImageFilePath varchar(255) = '',
    @sMenuItem1Desc varchar(200) = '',
    @cMenuItem1LargePrice udtMoney = 0,
    @sMenuItem2ImageFilePath varchar(255) = '',
    @sMenuItem2Desc varchar(200) = '',
    @cMenuItem2LargePrice udtMoney = 0,
    @sMenuItem3ImageFilePath varchar(255) = '',
    @sMenuItem3Desc varchar(200) = '',
    @cMenuItem3LargePrice udtMoney = 0,
    @sMenuItem4ImageFilePath varchar(255) = '',
    @sMenuItem4Desc varchar(200) = '',
    @cMenuItem4LargePrice udtMoney = 0,
    @sMenuItem5ImageFilePath varchar(255) = '',
    @sMenuItem5Desc varchar(200) = '',
    @cMenuItem5LargePrice udtMoney = 0,
    @sMenuItem6ImageFilePath varchar(255) = '',
    @sMenuItem6Desc varchar(200) = '',
    @cMenuItem6LargePrice udtMoney = 0,
    @sMenuItemPageFooter varchar(255) = '',
    @sStatus udtStatus = 'INACTIVE'

```

This page needs image upload and image change functionality for menu items similar in functionality to the Update Web site Home Page. The currently selected images should always be displayed.

2.2.1.8 UPDATE CATALOGUES LIST

The catalogue pages for this web site will be presented for selection via a results grid of some sorts. It should look and behave similar to the current result's grids in the system. When a user selects a catalogue page record from the grid, its fields will be displayed below the grid for selection.

See NetseekPhase2CataloguePageLayout.Doc for format.

These fields are to be displayed in the Catalogue list:

Field Name	Source	Heading Label	Functionality Required
ScatalogueCategoryName	tblCompanyCataloguePage	Catalogue Category	Ability to click and select. Clicking on this will show that record for editing below.
sCatalogueName	tblCompanyCataloguePage	Catalogue Title	Ability to click and select. Clicking on this will show that record for editing below.
mCatalogueDesc	tblCompanyCataloguePage	Description	
SStatus	tblCompanyCataloguePage	Page Status	

The following stored procedure will be called for retrieving results:

```

spgCatalogueList
    @lCompanyID int,
    @sPortalCode udtPortalCode = Null

```

If no sPortalCode is supplied, then the resultset will be filtered where bAllowWrite = 1

The resultset that is returned is as follows:

```

lCatalogueID,
sCatalogueCategoryName,
sCatalogueName,
mCatalogueDesc,
sStatus

```

Once a catalogue page record is selected, the catalogue edit page should be displayed for the selected record, looking similar to the existing catalogue pages, with the common fields presented in both the same vertical and horizontal order. All catalogue images are to be displayed as well.

Field Name	Source	Form Label Name	Functionality Required
ScatalogueCategoryName	TblCompanyCataloguePage	Catalogue Category	
sCatalogueName	tblCompanyCataloguePage	Catalogue Title	
mCatalogueDesc	tblCompanyCataloguePage	Description	
sCatalogueImageFilePath1	tblCompanyCataloguePage		This is the full path of the picture file to be displayed for the catalogue item. It should be displayed as an image. When the user selects to change an image, the system should concatenate the appropriate path string including the name of the file, and save it to this field,

			although the image should be displayed.
sCatalogueImageFileDesc1	tblCompanyMenu	Description	

These fields are then displayed for each of the 6 catalogue page items, which are sourced from the tblCataloguePageItem table.

For Each Catalogue Item:

Field Name	Source	Form Label Name	Functionality Required
iPageItemOrder	tblCompanyCataloguePageItem	Catalogue Item Order	Numeric. Allow values 1 to 6. Only digits.
sCatalogueItemDesc	tblCompanyCataloguePageItem	Item Description	
sCatalogueItemGroupName	tblCompanyCataloguePageItem	Item Type	This is a pick list and will be populated by the values in tblCatalogueItemGroup.CatalogueItemGroupName – In ascending order.
sCatalogueItemImageFilePath1	tblCompanyCataloguePageItem		This is the full path of the picture file to be displayed for the catalogue item. It should be displayed as an image. When the user selects to change an image, the system should concatenate the appropriate path string including the name of the file, and save it to this field, although the image should be displayed.
sCataloguePageFooter	tblCompanyCataloguePageItem	Menu Page Footer	
Sstatus	tblCompanyCataloguePageItem	Page Status	This field is a pick list. The options are: INACTIVE ACTIVE

The following stored procedure will be called for retrieving results:

```
spgCatalogueDetail
    @lCatalogueID int
```

The resultset that is returned is as follows:

```
tblCompanyCataloguePage.lCatalogueID,
tblCompanyCataloguePage.sCatalogueCategoryName,
tblCompanyCataloguePage.sCatalogueName,
tblCompanyCataloguePage.mCatalogueDesc,
tblCompanyCataloguePage.sCataloguePageFooter,
tblCompanyCataloguePage.sCatalogueItemImageFilePath1,
tblCompanyCataloguePage.sCatalogueImageFileDesc1,
tblCompanyCataloguePage.sStatus,

tblCataloguePageItem.lCatalogueItemID,
tblCataloguePageItem.iOrderSeed,
tblCataloguePageItem.sCatalogueItemDesc,
```

```
tblCataloguePageItem.sCatalogueItemGroupName,
tblCataloguePageItem.sCatalogueItemImageFilePath1,
tblCataloguePageItem.sStatus
```

The following stored procedures will be called for updating results:

```
spuCataloguePage
    @lCatalogueID int,
    @sCatalogueCategoryName char(19) = '',
    @sCatalogueName varchar(50) = '',
    @mCatalogueDesc varchar(4000) = '',
    @sCataloguePageFooter varchar(255) = '',
    @sCatalogueItemImageFilePath1 varchar(255) = '',
    @sCatalogueItemImageDesc1 varchar(50) = '',
    @sStatus udtStatus = 'INACTIVE'

and

spuCatalogueItem
    @lCatalogueItemID int,
    @iOrderSeed smallint = 6,
    @sCatalogueItemDesc varchar(255) = '',
    @sCatalogueItemGroupName char(20) = '',
    @sCatalogueItemImageFilePath1 varchar(255) = '',
    @sStatus udtStatus = 'INACTIVE'
```

This page needs image upload and image change functionality for menu items similar in functionality to the Update Web site Home Page. The currently selected images should always be displayed.

2.2.1.9 UPDATE DEAL LIST

The deal pages for this restaurant will be presented for selection via a results grid of some sorts. It should look and behave similar to the current result's grids in the system. When a user selects a special deal record from the grid, its fields will be displayed below the grid for editing.

These fields are to be displayed in the Deals list:

Field Name	Source	Heading Label	Functionality Required
sDealName	tblMenuDeal	Deal Name	Ability to click and select. Clicking on this will show that record for editing below.
sDealDesc	tblMenuDeal	Description	Ability to click and select. Clicking on this will show that record for editing below.
sStatus	tblMenuDeal	Deal Status	

The following stored procedure will be called for retrieving results:

```
spgMenuDealList
    @lCompanyID int,
    @sPortalCode udtPortalCode = Null
```

If no sPortalCode is supplied, then the resultset will be filtered where bAllowWrite = 1

The resultset that is returned is as follows:

```
tblMenuDeal.lDealID,
tblMenuDeal.sDealName,
tblMenuDeal.sDealDesc,
tblMenuDeal.sStatus
```

Once an announcement page record is selected, I'd like all fields to be displayed in a table containing three columns, as follows:

1. Name – Left aligned
2. Actual Field – Left aligned
3. Help – Left aligned – To contain text regarding the use of the fields.

The physical form length of each field should vary to reflect the number of characters allowed.

All fields and fonts are to default to the Arial Standard font.

Field Name	Source	Form Label Name	Functionality Required
sDealName	tblMenuDeal	Deal Name	
sDealDesc	tblMenuDeal	Description	
cDealPrice	tblMenuDeal	Total Deal Price	
sDealFooter	tblMenuDeal	Deal Footer	
sDealImageFilePath1	tblMenuDeal		This is the full path of the picture file to be displayed for the deal item. It should be displayed as an image. When the user selects to change an image, the system should concatenate the appropriate path string including the name of the file, and save it to this field, although the image should be displayed.
sStatus	tblMenuDeal	Page Status	This field is a pick list. The options are: INACTIVE ACTIVE

The following stored procedure will be called for retrieving results:

```
spgMenuDealDetail
    @lDealID int
```

The resultset that is returned is as follows:

```
tblMenuDeal.lDealID,
tblMenuDeal.iOrderSeed,
tblMenuDeal.sDealName,
tblMenuDeal.sDealDesc,
tblMenuDeal.cDealPrice,
tblMenuDeal.sDealFooter,
tblMenuDeal.sDealImageFilePath1,
tblMenuDeal.sStatus
```

The following stored procedure will be called for updating results:

```
spuMenuDeal
```

```

@lDealID int,
@iOrderSeed udtOrderSeed = 99,
@sDealName varchar(50) = '',
@sDealDesc varchar(255) = '',
@cDealPrice udtMoney = 0,
@sDealFooter varchar(255) = '',
@sDealImagePath1 varchar(255) = '',
@sStatus udtStatus = 'INACTIVE'

```

This page needs image upload and image change functionality like the Update Web site Home Page. The currently selected image should always be displayed.

2.2.1.10 UPDATE USER DETAILS

Once selected the following fields will be displayed in a table containing three columns, as follows:

1. Name – Left aligned
2. Actual Field – Left aligned
3. Help – Left aligned – To contain text regarding the use of the fields.

The physical form length of each field should vary to reflect the number of characters allowed.

All fields and fonts are to default to the Arial Standard font.

Field Name	Source	Form Label Name	Functionality Required
sUserID	tblCompany	User ID	Read Only – Cannot change this. It will be system assigned eventually.
sUserTitle	tblCompany	Title	Pick list containing 'Mr, Ms, Mrs, Miss, Dr, Prof.'
sUserFName	tblCompany	First Name	
sUserSName	tblCompany	Surname	
sUserPhone	tblCompany	Phone	
sUserFax	tblCompany	Fax	
sUserEmail	tblCompany	Email Address	
sUserPassword	tblCompany	Login Password	Force entry twice for validation. Display each letter as an '*'

The following stored procedure will be called for retrieving results:

```

spgUserDetail
    @lCompanyID int

```

The resultset that is returned is as follows:

```

lCompanyID,
sUserID,
sUserTitle,
sUserFName,
sUserSName,
sUserPhone,
sUserFax,
sUserEmail,
sUserPassword

```

The following stored procedure will be called for updating results:

```

spuUserDetail
    @lCompanyID int,
    @sUserID char(15),
    @sUserTitle varchar(5) = '',
    @sUserFName varchar(25) = '',
    @sUserSName varchar(35) = '',
    @sUserPhone udtPhone = '',
    @sUserFax udtPhone = '',
    @sUserEmail varchar(50) = '',
    @sUserPassword varchar(10) = ''

```

This page needs image upload and image change functionality like the Update Web site Home Page. The currently selected image should always be displayed.

2.2.1.11 STATISTICS

The statistics page will prompt the user for the date range to display the statistics for. The following fields will be displayed based on the results returned from spgPageStatistics.

All fields and fonts are to default to the Arial Standard font.

Field Name	Source	Form Label Name	Functionality Required
tFrom	Entered Variable	From Date	Enter the From Date for statistics range – Display at top of page.
tTo	Entered Variable	To Date	Enter the To Date for statistics range – Display at top of page.
SType	spgPageStatistics	Page Type	R/O – Displayed In Table
SPageName	spgPageStatistics	Description	R/O – Displayed In Table
lvisits	SpgPageStatistics	Page Views	R/O – Displayed In Table

A new stored procedure exists that displays a list of page 'hit' statistics

The following stored procedure will be called for retrieving results:

```

spgPageStatistics
    @sPortalCode udtPortalCode = 'Netseek',
    @tFrom datetime,
    @tTo datetime

```

Assumption: That only dates are passed to the procedure (not datetimes), therefore the time component will be 00:00:00.

Eg: Passing @tFrom = '01/01/2000' and @tTo = '01/01/2000' will filter records that are >= '01/01/2000 00:00:00' And <= '01/01/2000 23:59:59'

The resultset that is returned is as follows

```

sType,
sPageName,
lvisits

```

where

sType is the Page Type

sPageName is the record's main descriptive field

lVisits is the count of 'visits' on that page

2.2.1.12 UPDATE LINKS LIST

The links pages for this web site will be presented for selection via a results grid of some sorts. It should look and behave similar to the current result's grids in the system. When a user selects a links list record from the grid, its fields will be displayed below the grid for editing.

These fields are to be displayed in the Links list:

Field Name	Source	Heading Label	Functionality Required
sNameOfLink	spgCompanyLinkLists	Name Of Link	
sURL	spgCompanyLinkLists	URL	

```
spgCompanyLinkList
    @lCompanyID int,
    @sPortalCode udtPortalCode = Null
```

If no sPortalCode is supplied, then the resultset will be filtered where bAllowWrite = 1

The resultset that is returned is as follows:

```
lCompanyLinkID,
sNameOfLink,
sURL
```

This stored procedure will retrieve a list of additional hyper links that the web site may set up to be displayed in the left hand side navigation bar of the .asp pages.

Once an link record is selected, I'd like all fields to be displayed in a table containing three columns, as follows:

1. Name – Left aligned
2. Actual Field – Left aligned
3. Help – Left aligned – To contain text regarding the use of the fields.

The physical form length of each field should vary to reflect the number of characters allowed.

All fields and fonts are to default to the Arial Standard font.

Field Name	Source	Form Label Name	Functionality Required
sNameOfLink	spgCompanyLinkDetail	Name Of Link	
sURL	spgCompanyLinkDetail	URL	This field will allow the user to select if they wish to reference an internal link or an external link. If its internal, the system will call

			<p><i>spgActiveLinks</i> @iCompanyID int</p> <p>(A new stored procedure exists that displays a list of all Active records within a web site. The user is able to select a record to become a hyper link that is stored in the tblCompanyLink table.)</p> <p>The resultset that is returned is as follows:</p> <p>'iOrder', 'iTableID', 'sType', 'sName', 'sDesc'</p> <p>where iOrder is</p> <p>1 for web site records, 2 for product/services records, 3 for catalogue page records, 4 for menu records, 5 for menu deal records and 6 for announcement records</p> <p>iTableID is the corresponding table's ID field value</p> <p>sType is the table name</p> <p>sName is the record's main descriptive field eg: sMenuDealName</p> <p>sDesc is the record's secondary descriptive field eg: sDealDesc</p> <p>If its an external link, the user can type in the URL themselves.</p>
bOpenInNewWindow	spgCompanyLinkDetail	Open New Browser	Yes or No picklist.
iOrderSeed	spgCompanyLinkDetail	Display Order	Allow numerics. 2 digits Max.

The following stored procedure will be called for retrieving results:

```
spgCompanyLinkDetail
    @iCompanyLinkID int
```

The resultset that is returned is as follows:

```
iCompanyLinkID,
sNameofLink,
sURL,
bOpenNewWindow,
iOrderSeed
```

The following stored procedure will be called for updating results:

```
spuCompanyLinkDetail
    @lCompanyLinkID int,
    @sNameofLink varchar(20) = '',
    @sURL varchar(150) = 'http://www.netseek.com.au',
    @bOpenNewWindow udtYesNo = 0,
    @iOrderSeed udtOrderSeed = 99
```

A new data export stored procedure exists. This is called by

```
spPortalExport
    @sPortalCode udtPortalCode,
    @iExportOption smallint = 0
```

The iExportOption can be one of the following:

- 0 – exports Portal data
- 1 – exports web site data
- 2 – exports product/services
- 3 – exports announcements
- 4 – exports menus
- 5 – exports menu deals
- 6 – exports catalogues
- 7 – exports web site links

2.2.1.13 PURCHASE A WEB SITE / ADDITIONAL PAGES

Netseek will allow users to purchase either a web site or additional web pages through a payment interface.

The transaction component will be outsource to a transaction house, who will host the pages that select the product being purchased, instigate the transaction, and on success, notify netseek that the transaction was approved.

On approval, a record needs to be inserted into spiWWWRequest.

A new stored procedure exists that will submit a request from the WWW for a Netseek package.

```
spiWWWRequest
    @lCompanyID          int = 0,
    @sPortalCode         varchar(10) = 'Netseek',
    @sCompanyName        varchar(50),
    @sContactTitle       varchar(5),
    @sContactFName       varchar(25),
    @sContactSName       varchar(35),

    @sinformBy           varchar(10) = 'EMAIL',
    @sContactEmail       varchar(50),
    @sContactFax         varchar(15),

    @iPackage            smallint = 0,

    @sTransactionNo      varchar(40),
```


@sReceiptNo *varchar(40),*
@tTransactionDate *datetime = CURRENT DATE*

Each particular package (@iPackage) will insert appropriate records into the tblRequest... tables.

A process will periodically scan this table for new entries and process the requests as required. Users are notified of their web-site creation, and are either emailed or faxed the username and password.

2.2.1.14 FORGOT PASSWORD

A field needs to be placed on the home page (or somewhere) which allows a user to enter their own email address. This email address is then passed to spSendUserPassword, which will send that email address its password.

The new stored procedure will send the company contact their UserID and password if they forgot it.

The stored procedure

spSendUserPassword
@sUserEmail varchar(50)

will be called from an .asp page and will write a record into the tblFaxEmail table. This table will later be scanned by an automated process to send the email.

Appendix C

Netseek Portal Creation Process Definition

1. Populating Netseek Database Tables – Create Portal

The functionality of populating the Netseek database to create a portal will comprise a component that will allow a user to purchase a portal package online.

1.1 Portal Creation Process

This ASP will allow an operator to create any number of records for certain portal creation tables, which translate into a portal entity within the database.

The first step allows entry of default information for creating new portals within the database.

1.1.1 Create New Portal

A series of data entry pages are presented for entering default portal website information, as follows.

1. Main Page
2. Content/Products/Services
3. Announcements
4. Catalogues and Catalogue Items
5. Menus
6. Deals

The user will be able to navigate through the series of screens entering default information and specifying how the portal websites should look, and the default template information it should contain.

When the CREATE button is pressed, the system will write the records to the tblRequestPortal tables for processing.

1.1.2 Creating Records

This forms the core of the population system. This creation procedure is called from the Creation Polling Routine and also when the user clicks the Create button to create a portal.

The actual portal creation process starts after the user selects the CREATE button after the portal template and data collection process.

This action calls the stored procedure spInsertManual to insert a blank record into each of the SQL tblRequest... tables and returns the IRequestID, which is unique to that specific portal.

The portal template information and data that is entered by the end user is then exported via update queries to the tblRequest records just created. The IRequestID is used to reference the correct portal creation records in the event that numerous users are creating portals simultaneously.

When the portal template information and data is written to the tblRequest tables, the procedure spCreateRecords is called, passing the current IRequestID as an identifier for the portal being created.

This process commences at the top of the tblRequestPortal, for each IRequestID, if a specific IRequestID is not passed.

1.1.3 Create the Actual Portal Website Records (IRequestID Specific)

The spCreateRecords process inspects details of the 'Create Portal' request, (bAddMorePages, IWebsiteID, iRequestQty) in the underlying tblRequest... tables and creates the table records via three sub procedures. These are spCreatePortal, spCreatePortalChild, and spCreateOther.

The spCreatePortal procedure creates the portal 'main page' records and also calls spCreateWebsiteChild and spCreateOther. This procedure accepts the IRequestID and iChildQty parameters so that it can process the correct request record and flag each Website as individual or parent.

The spCreateWebsiteChild procedure solely creates Website records that are flagged as children and attaches them to a parent. This procedure accepts the IRequestID and IWebsiteID parameters so that it can process the correct request record and attach it to the correct Website.

The spCreateOther procedure will create all other Website related records (Announcement, Menu, MenuDeal, ProductService, CataloguePage, and CatalogueItem) and attach them to a Website record. This procedure accepts the IRequestID and IWebsiteID parameters so that it can process the correct request record and attach it to the correct Website.

The password is generated as a string of random letters and digits in a new stored procedure called spCreatePassword. This procedure is called within the spCreatePortal, and spCreateWebsiteChild procedures.

This process loops to create the next portal website, until the requested number of websites are created within the portal.

After the records are successfully created, the request records that match the passed IRequestID are then deleted from the tblRequest tables.

1.1.4 spCreateRecords

A scheduled job exists on the Portal Server that will periodically scan the tblRequestWebsiteParent table for records that are flagged as WWW requests (sRequestSource = 'W') and that have the transaction approved (bTransApproved = True).

This routine will loop through all the pending transactions that match the criteria and call the Record Creation procedure for each request (see 1.1.3). If the Record Creation procedure returns with a Success status, then the user is informed that the request has been processed and will email them with their UserID and Password.

It is expected that during the processing of a portals creation, this polling routing will be processing hundreds of records at a time.

1.4 Fax/Email Requestor – To Notify That Portal and Websites

When records have been created via the WWW, a record is to be written to the tblFaxEmail table so that either a Fax or Email can be sent to the user to inform them that their portal has been created.

If an individual website is created, the email or fax should be sent immediately (Status = 'PENDING').

The procedures spCreateRecords and spCreatePortal call a procedure called spiEmail, which writes a record to the tblFaxEmail table. An external process then interrogates this table and sends the faxes/emails. See 1.5 below.

1.5 Fax/Email Process

This process retrieves all 'PENDING' items in the tblFaxEmail table.

The process opens this resultset and inspects the sCorrespondanceType field. If it is 'FAX', then a fax file is and sent to the fax server. If it is 'EMAIL', then an email is sent via the ASPMail. If ASPMail detects an error in sending the email, the error message is written to the sFailReason field in the tblFaxEmail table.

The emailing component has two new entries in the NetSeekFax.ini file. These are:

[SMTP SERVER], which specifies the email server address (eg: mail.netseek.com.au)

and

[SMTP TIMEOUT], which specifies how long ASPmail should attempt to send the email in seconds.

After the resultset has been processed, the process will poll the working folder (specified by the [FAX/EMAIL PROCESS PATH] entry in the NetseekFax.ini file) every minute to determine if all fax files have been processed by GFIFaxMaker. When no files exist with the extension .txt (all have been processed by GFIFaxMaker), the fax files are organised into Success and Error subfolders and the program is closed.

An NT Schedule has been set up to start the polling every 15 minutes. More than one process session is not be allowed. The process should check to see if it is already running on activation.

Appendix D

Search Architecture ASP Pages

Portal Entry page.

Starting point into a portal search engine is a single page (can be static or dynamic). This is a content providing page containing the portal code together with basic search criteria: keywords, suburb, region, state. These are embedded in the form tags as an input text.

Control is passed to seek.asp using POST method. Action selected is embedded in the image selected:

depending on the selection like `<... name=announcement src="images/purple_01_02.gif" type=image width=55>`.

There are 4 types of actions: catalogue, announcement, company and menu. The Portal code for this page is hardcoded into it. The main portal search page has a Portal code of 'that' corresponds to a value in the tblPortalCode table.

Dynamic pages

Dynamic pages (ASP) are located in /scripts directory. Structure of most of the dynamic pages is similar. For modularity and extensibility certain functions are located in the server side includes. Here is an example of the seek.asp includes:

```
<!--#include file = "../scripts/style.css"-->
[common style elements of the site like hyperlink colors, table cells colors, etc]

<!--#include file = "aspconst.inc"-->
[ASP constants definition - optional]

<!--#include file = "commonvb.inc"-->
[Common functions used throughout the system like setting stored procedure parameters, building search bar, etc]

<!--#include file = "commonvbseek.inc"-->
[Page specific functions. This varies from page to page. Current page specific includes:

commonvbdetl.inc
commonvbdetl_catalogue.inc
commonvb enrol.inc
commonvbmsg.inc
commonvbseek.inc]
...
<!--#include file = "../scripts/footer.inc"-->
[Dynamically creates page footer with Copyright information]
```

Most of the information on the dynamic pages comes from the database (delivered through stored procedures and populated to HTML tags through ASP functions).

Common programming mechanism for getting this information looks like:

```
Set Conn = Server.CreateObject("ADODB.Connection")
'/spgqmCompany
Set RS = Server.CreateObject("ADODB.Recordset")
Conn.Open "NETSEEK", "SQLuser", "password"

Set cmd = server.createObject("ADODB.command")
set cmd.activeconnection = Conn
```

```
cmd.CommandType = adCmdStoredProc
sType="Company"
SetSPParamsDetl cmd, sID, sType
Set RS = cmd.Execute
```

Database data is displayed in two ways. Either through direct insertion of the HTML tags like:

```
<font size="2"><%=RS("sProdServPageFooter")%></font>
```

or through HTML formatting subfunctions like:

```
<%=GetProdServBar(cmd, sID, RS1, RS) %>
<%=GetCompanyHeader(RS) %>
<%=GetProdServBar(cmd, sID, RS1, RS) %>
<%= GetCompanyInfo(RS, RS2, "Product")%>
```

Seek page.

Seek page is preferably a recursive dynamic page. It's preferably called in 3 scenarios: from the static page like a portal entry page(above), a new search from company pages or by itself (when displaying search results). When called by entry page or new search option it sets itself in corresponding mode (business, announcement, catalogue or restaurant). When traveling through Netseek system user preferably always stays in one of those modes until new search is initiated from the entry page. Seek uses server site variables to preserve information between the searches or throughout result navigation.

To identify users and match them with server side temporary storage Netseek sites use temporary cookies. Users cookie holds a unique session handle that allows server side ASP to identify which information to restore.

Initial search is preferably always executed using POST method. Navigation through the results is preferably achieved through GET method.

Here is a list of dynamic pages with the basic description:

File Name	Function/Description
1prodserv.asp	Retrieves specific product, service or content information. Executed when user selects particular product, services or content links displayed on the left hand side bar (in main page area)
2menu.asp	Retrieves specific menu information. Executed when user selects particular menu link displayed on the left hand side bar (in company area)
3deal.asp	Retrieves specific special offer information. Executed when user selects particular special offer link displayed on the left hand side bar (in company area)
announcement.asp	Retrieves specific announcement information. Executed when user selects particular announcement from the announcements grid.
announcements.asp	Retrieves all the announcements available for this parent. Results are presented in grid format.
catalogue.asp	Retrieves specific catalogue information. Executed when user selects particular catalogue from the announcements grid.
catalogues.asp	Retrieves all the catalogues available for this company. Results are presented in grid format
company.asp	Retrieves detailed information for the specific company and home page.

contact.asp	Retrieves basic company information (like principle, etc) and builds contact company page. Processing is passed to sndmessage.asp (POST)
enrolbasic.asp	Allows user to enter basic subscription information. Processing is passed to enrolbasicend.asp (POS)
enrolbasicend.asp	Processes enrolment information. Creates new entries in the database.
seek.asp	Main search functionality.
sndmessage.asp	Processes contact information. Inserts basic info and message into the database for faxing or emailing.

Portal Files

ROOT:

global.asa
default.htm

ROOT\images:

background.jpg
background2.jpg
Banner.GIF
BANNER.jpg
bannereg.gif
btnannouncement.jpg
btncatalogue.jpg
btncompany.jpg
btnmenu.jpg
clear.gif
ColourLogoSmall.jpg
ColourLogoSmallEmboss.jpg
interimhome.gif
keywords.gif
lilacbackground.jpg
nslogo.jpg
purple_00.gif
purple_01_01.gif
purple_01_02.gif
purple_01_04.gif
purple_01_05.gif
purple_02_02.gif
purple_02_03.gif
purple_03_03.gif
purple_04_02.gif
purple_04_04.gif
purple_05_01.gif
purplebutton.gif
region.gif
smlogo.jpg
smlogo1.jpg
state.gif
suburb.gif
thinblueline.gif
Thinblueline_hp.gif

ROOT\scripts:

1prodserv.asp
2menu.asp
3deal.asp
announcement.asp
announcements.asp
aspconst.inc
catalogue.asp
catalogues.asp
commonvb.inc
commonvbdetl.inc
commonvbdetl_catalogue.inc
commonvbenrol.inc
commonvbmsg.inc
commonvbseek.inc
company.asp
contact.asp
enrolbasic.asp
enrolbasicend.asp
footer.inc
seek.asp
sndmessage.asp
style.css
style.inc

Appendix E

Portal Creation Processes and Functions

Version: 7.3

1. Populating Netseek database

The functionality of populating the Netseek database to create a portal will comprise a component that will allow a user to purchase a portal package online.

1.1 Automatic Portal and Website Creation and Population (via WWW)

1.1.1 Information Collection

Any Internet user will have the option to visit a Purchase page on the site and purchase any one of a set of pre-defined packages.

The first page presented to the user will consist of the collection of mandatory information.

The user will be asked to enter information about their company such as Name, Address, and Contact details. The user will also be informed when their portal request has been completed, by e-mail.

1.1.2 Purchase Portal Pages

The next page that is presented will allow the user to select one of many predefined portal or website packages that are available. A portal package may exist that will ask the user how many websites to create within the portal. A website package may exist specific to restaurants and cafes or specific to a retailer.

Eg:

An example of a portal might be a shopping mall containing 150 shops, each requiring 4 content pages, and 10 catalogue pages, with shopping cart functionality. Netseek would create 150 records in the tblCompany table through this process.

A restaurant may purchase the following package:

1 home page + 3 Menu pages (each menu containing 6 items)

Whereas a hardware store may purchase:

1 home page + 2 Announcements
+ 5 Catalogues (each catalogue with 6 items)

The user will also be asked to enter their credit card details for the online purchase.

1.1.3 Request Processing

Initial Requirement

A stored procedure will be called when the user submits the requests. The stored procedure will accept as parameters all the portal and company specific fields that they entered as well as an integer flag to indicate what package they have selected.

This flag will be inspected and will create records in the tblRequest... tables that correspond to the package indicated. In the tblRequestCompanyParent table, the sRequestSource flag will be set to 'W' and the bTransApproved flag set to False.

The Transaction Server will then attempt to make the purchase via the entered credit card details. If successful, the bTransApproved flag will be set to True to indicate that the purchase was successful and that the polling routine can process the request.

What was implemented

A new stored procedure has been created called spiWWWRequest. This procedure accepts key company fields such as sCompanyName, sContactName etc... and also an integer to indicate what package they have selected.

The procedure then creates the corresponding records in the tblRequest... tables according to the selected package. The polling routine will then extract this information from the tblRequest... tables and process the requested records.

1.1.4 Polling Routine

Initial Requirement

A scheduled job will be set up on the Netseek SQL Server that will periodically scan the tblRequestCompanyParent table for records that are flagged as WWW requests (sRequestSource = 'W') and that have the transaction approved (bTransApproved = True).

This routine will loop through all the pending transactions that match this criteria and call the Record Creation stored procedure for each request (see 1.3). If the Record Creation stored procedure returns with a Success status, then the user is informed that the request has been processed and will either fax or email them with their UserID and password.

It is expected that if a portal is requested, this polling routine will be processing hundreds of records at a time.

1.2 Manual Population (via Access ASP)

This ASP will allow an operator to create any number of records for certain tables, which translate into a portal.

The first option will allow entry of default information for creating new portals within the Netseek database.

1.2.1 Create New Portal

A series of screens will be presented for entering default portal website information. The screens that will be presented are:

- + Company
- + Content/Products/Services
- + Announcements
- + Catalogues and Catalogue Items
- + Menus
- + Menu Deals

The user will be able to navigate through the series of screens entering default information and specifying how the portal websites should look, and the default information they should contain.

When the create button is pressed, the system will write the records to the WWWRequest table for processing.

1.2.2 Creating Records

The actual record creation process starts after the user clicks the Create button that is on the last data entry screen. This action will call the stored procedure `spInsertManual` to insert a blank record into each of the SQL `tblRequest...` tables and return the `lRequestID`.

The data that is temporarily stored will then be exported via Update queries to the records just created on SQL server (using the `lRequestID` to update the correct ones).

When the data is in SQL Server, the procedure `spCreateRecords` is called. This procedure accepts the `lRequestID` to inform it what records it should process.

1.3 Create Records Process

Initial Requirement

This will form the core of the population system. This stored procedure will be called from the Polling Routine and when the user clicks the Create button to create a portal.

This process commences at the top of the file, for each `lRequestID`

This procedure will accept the `lRequestID` as a parameter to indicate what record from the request tables should be processed as many requests may exist in these tables.

The entire process will be enclosed within a transaction so should something fail, nothing will get written to the 'Live' database.

Records will be extracted from each table that match the passed `lRequestID` and the corresponding record will be written to the 'Live' tables.

If the procedure is processing a manual request ('M'), the `bNewRecords` flag will be inspected to determine if new or more records are to be created and will process the request accordingly.

At this stage, the `UserID` will be generated. The `UserID` must be unique and will be generated via the following algorithm:

`UserID` = First three left most characters from the selected `PortalCode` + `lCompanyID` field

Eq: `UserID` = 'NET96832015' or 'WES87642389'

The Password for the user will always be set to 'password'.

If the entire process of creation is successful, then the records in the request tables are deleted for that particular `lRequestID`.

This process must also create a string of the `UserID` and passwords that were created. This string is then to be written to the `tblFaxEmail` table so that the Fax/Email Process can process the requests and send off a corresponding fax or email.

Actual Functionality

The `spCreateRecords` procedure inspects details of the Request (`bAddMorePages`, `lCompanyID`, `iRequestQty`) in the underlying `tblRequest...` tables and creates the according records via three sub procedures. These are `spCreateCompany`, `spCreateCompanyChild`, and `spCreateOther`.

The `spCreateCompany` procedure creates company records and also calls `spCreateCompanyChild` and `spCreateOther`. This procedure accepts the `lRequestID` and

iChildQty parameters so that it can process the correct request record and flag the company as individual or parent.

The spCreateCompanyChild procedure solely creates company records that are flagged as children and attaches them to a parent. This procedure accepts the IRequestID and ICompanyID parameters so that it can process the correct request record and attach it to the correct company.

The spCreateOther procedure will create all other company related records (Announcement, Menu, MenuDeal, ProductService, CataloguePage, and CatalogueItem) and attach them to a company record. This procedure accepts the IRequestID and ICompanyID parameters so that it can process the correct request record and attach it to the correct company.

The password does not default to 'password'. It is generated as a string of random letters and digits in a new stored procedure called spCreatePassword. This procedure is called within the spCreateCompany, and spCreateCompanyChild procedures.

This process now loops to create the next portal website.

After the records are successfully created, the request records that match the passed IRequestID are then deleted from the tblRequest... tables.

1.4 Inform Fax/Email Requestor that pages have been created

Initial Requirement

When records have been created via the WWW, a record is to be written to the tblFaxEmail table so that either a Fax or Email can be sent to the user to inform them that their portal has been created.

If an individual website is created, the email or fax should be sent immediately (Status = 'PENDING').

The procedures spCreateRecords and spCreateCompany call a new stored procedure called spiEmail. This procedure will write a record to the tblFaxEmail table to notify the requestor that the records have been created.

12b. Display Active Links

Initial Requirement

When the user is editing their home page details, a section will be displayed on the .asp page to allow a link to be set up that points to a particular page within their Netseek site or to an external site. If the user has chosen to create links to a particular page within their Netseek site then a list of all 'Active' pages are displayed.

From this list, the user will be able to select a page that is to have a link displayed on the left hand portion of the .asp page when viewing their portal website. This functionality is purely for the purpose of showcasing specific pages with links from the main page.

A stored procedure will be created called spgActiveLinks that will return the following fields:

The ID of the particular table, the Type of page (ie: from which table), the page name, and the page description. All records returned will be filtered on 'Active' and will be returned in the order of home page, product/services, catalogues, menus, menudeals and announcements.

Eg:

ID	Type	Name	Desc
1	home page	Acme repair service	We fix all electrical items
2	prod/serv models	repairs	Repairs done on all makes and
1	catalogue	Acme catalogue	Acme April Specials Catalogue

A new stored procedure, spgActiveLinks, returns a resultset of all the 'Active' records within a company (company record, tblCompanyProdServ, tblCompanyCataloguePage, tblCompanyMenu, tblMenuDeal, and tblCompanyAnnounce) and returning them as a single resultset as outlined in the initial requirement.

9. Remove Invalid Characters

Initial Requirement

The search stored procedures are using the CONTAINS predicate to search on Keywords. If the keywords contain invalid characters (characters that are non alphanumeric), then the keyword search may fail.

To prevent this from occurring, a generic procedure will be constructed to scan the keyword string and remove any invalid character(s).

This procedure will be called by all four of the 'search engine processors' prior to extracting individual keywords, and will work as follows:

For each character in the keyword string

If the character is not numeric (0 to 9) or not alphabetic (a to z and A to Z), then it is an invalid character.

If the character is a space, then ignore it because the space character delimits the keywords in the string.

Replace the invalid character with a space character (effectively removing the invalid character and separating words)

loop

Pass the new string to the procedure for extracting individual words.

Eg: "bargain,home & shopping" will be converted to
"bargain home shopping"

This string will then have the keywords extracted from it:

"bargain"
"home"
"shopping"

A new stored procedure has been created called spValidateKeyword. This procedure is called from all four search stored procedures when either the 'all the words' or 'any of the words' querytype is specified.

15c. Retrieve statistics

Initial Requirement

Portal's owners may want to retrieve statistical data about their portals visits and activity. This will be achieved by creation of a new stored procedure that accepts a portalCode and a Start and End date range.

The procedure will count the number of records in the tblStatsCompanyPage table for the Portal that are within the Start and End dates inclusive.

A resultset will be returned with one field that will record the number of visits. Records will be returned in the following order:

1. Company
2. Announcement
3. Menu
4. Menu Deal
5. Product/Service
6. Catalogue

A new stored procedure has been created called spgPageStatistics. This procedure assumes that only dates (not date/times) will be passed to the procedure, therefore the time component will always be 00:00:00.

Three fields are returned, Type, Page Name, Number of Visits. The resultset is sorted by the Type (which is in the order above) and then by the number of visits in descending order.

10. Automatic Activation/Deactivation routine**Initial Requirement**

A scheduled process will be set up on the SQL Server to periodically scan through the tables and activate and deactivate records.

A stored procedure will be run called spActivateDeactivate that will scan the following tables:

tblPortal
tblCompany
tblCompanyAnnounce
tblCompanyMenu
tblMenuDeal
tblCompanyProdServ
tblCompanyCataloguePage (includes tblCataloguePageItem)

Two updates will occur on each of the above tables:

- 1) Activate inactive records
Update the sStatus field to 'Active' if the current system datetime falls within the record's start and end dates and if the sStatus field is 'Inactive'
- 2) Deactivate active records
Update the sStatus field to 'Inactive' if the current system datetime falls outside of the record's start and end dates and if the sStatus field is 'Active'
- 3) Website home page records will be excluded from this process.

Created a new stored procedure (spActivateDeactivate) that performs the above. This procedure is called by a SQL Server job called Activate/Deactivate records.

11. Email/Fax customer of due expiration

Initial Requirement

The Netseek customer is to be informed if any of his/her pages are due for expiration within a pre-determined time period. A scheduled SQL Server job will be set up to scan the following tables and determine if any of the records are due for expiration:

tblPortal
tblCompany
tblCompanyAnnounce
tblCompanyMenu
tblMenuDeal
tblCompanyProdServ
tblCompanyCataloguePage

This routine will call a stored procedure that calculates a date in the future (calculated by the current date + a set interval) and determine if this calculated date exceeds the EndDate of any records in the above mentioned tables.

One message will be sent that will contain details of which item is due to expire.

The customer will be either faxed or sent a notice that the item(s) is about to expire soon and that payment must be made to reactivate the item.

The procedure is to inform the user at 28 days, 14 days, 7 days and at 0 days. At 0 days – the website is Inactive – inform user that the record has been deactivated.

A new stored procedure has been created called spInformExpiry that is run by a scheduled SQL Server job every 24 hours.

A period can be set up in the procedure to determine when a record should be classed as about to expire. Currently this period has been set up as 14 days in the future.

The procedure will scan the above tables and retrieve all records that are Active and have an EndDate that falls on the calculated 'cut off' date (14 days from now).

These records are inserted into a single message that is written into the tblFaxEmail table. These messages will always be emailed.

The procedure won't pick up the records again that next time it is executed.

20. Administration Interface

20.1 Masterfiles

Clicking this option will open the Masterfiles submenu that will contain a screen for editing each of the following tables:

tblPortal
tblAnnounceCategory
tblBusinessCategory
tblCatalogueCategory
tblCatalogueItemGroup
tblIgnoredWord
tblMenuStyle
tblProdServCategory

20.2 Company Details

Clicking this option will display a company search screen. A search must be performed to retrieve one company. This is to ensure that the speed is optimal when editing a large database.

Once a company for editing is selected, a submenu with the following options is displayed each of which will open a screen to edit the corresponding table:

- tblCompany
- tblCompanyAnnounce
- tblCompanyBanner
- tblCompanyCataloguePage (includes tblCataloguePageItem)
- tblCompanyMenu
- tblCompanyPayment
- tblCompanyProdServ
- tblFaxEmail
- tblMenuDeal

20.3 System Tables

This menu option accesses all system tables that the user would not normally be aware of. Clicking this option will open the System Tables submenu that will contain a screen for editing each of the following tables:

- tblRequestAnnouncement
- tblRequestCataloguePage (includes tblRequestCatalogueItem)
- tblRequestCompanyChild
- tblRequestCompanyParent
- tblRequestMenu
- tblRequestMenuDeal
- tblRequestProductService
- tblStatsCompanyPage
- tblStatsNetseekPage

23. Export Portal Data

Initial Requirement

The Portal's owners may need to view a listing of all the records that they have in their portal. An .asp page will be created that allows the Portal owner to log on and select one of many options that export data from a particular table (eg: Export companies or Export Product/Services)

The options for export are as follows:

- 0) tblPortal
- 1) tblCompany
- 2) tblCompanyProdServ
- 3) tblCompanyAnnounce
- 4) tblCompanyMenu
- 5) tblMenuDeal
- 6) tblCompanyCataloguePage (this will include CatalogueItems)
- 7) tblCompanyLink

A stored procedure will be created that accepts sPortalCode and iExportOption as parameters. Each option will export most fields from the table – only fields that are specific to

Netseek will be excluded, such as ICompanyID, bVisibleToNetseek (as these have no meaning for the Portal).

28. Login Stored Procedure

Initial Requirement

Develop a login stored procedure that accepts a UserID and password. This procedure will first try to locate the User in the Portal table and if not found, then it will try and find it in the tblCompany table.

If not found in either table, then a resultset will be returned outlining the error (Greg will reply to this), otherwise, the following fields will be returned:

sType_varchar(1)

This field indicates if the user belongs to an (A)ffiliate, (I)ndividual Company, (P)arent Company or (C)hild Company.

sPortalCode

This field will store the PortalCode of the user if the Type is 'A'.

ICompanyID

This field will store the ICompanyID of the user if the Type is 'I', 'P', or 'C'.

Created procedure spvValidateUser that accepts sUserID and sPassword. The procedure will always return bSuccess as the first field that is either 1 for success or 0 for failure.

If the user is valid, the fields sType, sPortalCode, and ICompanyID are returned after bSuccess.

If the user is invalid, the field sReason is returned after bSuccess.

32. Email UserID + Password

Initial Requirement

The WWW page will allow a user to enter their email address so that their UserID and Password can be emailed to them if they forgot it.

The stored procedure will accept an email address and attempt to find it in the tblCompany table on the sUserEmail field. If not found, an error will be returned, otherwise the UserID and Password are written to the tblFaxEmail table and sent to the sContactEmail address.

What was Implemented

A new stored procedure has been created called spSendUserPassword. It accepts the sUserEmail as a parameter and attempts to locate it in the tblCompany table. If found, a record is written to the tblFaxEmail that is directed to the email address of the Contact of the Company.

33. Fax/Email Process

This process retrieves all 'PENDING' items in the tblFaxEmail table.

The process opens this resultset and inspects the sCorrespondenceType field. If it is 'FAX', then a fax file is sent to the fax server. If it is 'EMAIL', then an email is sent via the ASPMail. If ASPMail detects an error in sending the email, the error message is written to the sFailReason field in the tblFaxEmail table.

The emailing component has two new entries in the NetSeekFax.ini file. These are:

[SMTP SERVER] which specifies the email server address (eg: mail.netseek.com.au)

and

[SMTP TIMEOUT] which specifies how long ASPmail should attempt to send the email in seconds.

After the resultset has been processed, the process will poll the working folder (specified by the [FAX/EMAIL PROCESS PATH] entry in the NetseekFax.ini file) every minute to determine if all fax files have been processed by GFIFaxMaker. When no files exist with the extension .txt (all have been processed by GFIFaxMaker), the fax files are organised into Success and Error subfolders and the program is closed.

An NT Schedule will be set up to start the polling every 15 minutes. More than one process session will not be allowed. The process should check to see if it is already running, on activation.

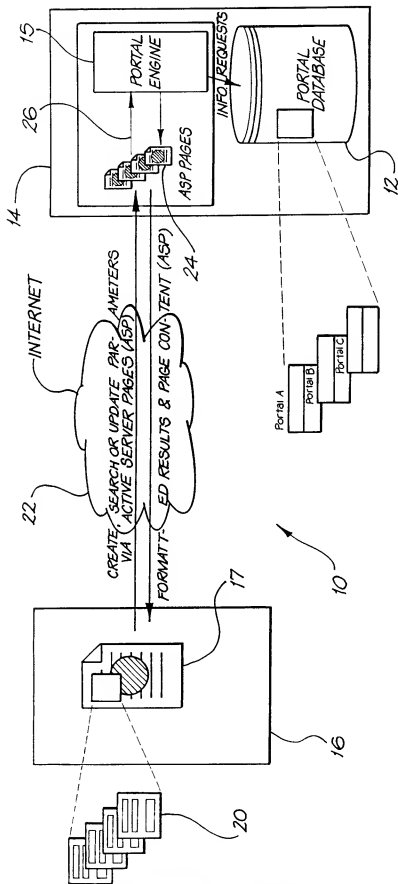


FIG. 1

Portal Creation Flow Chart

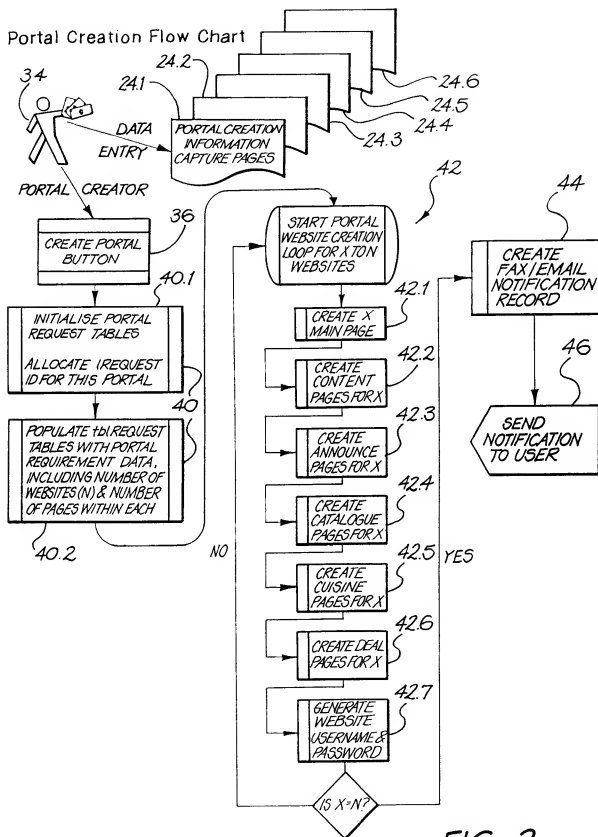


FIG. 2

Portal Website Number & Web Site Details	
Number Of Websites	10
Portal Code	CHAMBER
Visible To Netseek:	Yes
Business Category	Automotive
Sub Type:	
Order Seed:	70
Number Of Pages:	5
Options	/chamber/images/logo.jpg
Logo Path:	<Enter Your Business Name>
Company Name	
Street1:	<Enter Your Street Name>
Street2	<Enter Your Street Name>
Suburb	<Enter Your Suburb>
Region:	Inner City
Post Code	0000
State	NSW
Main Phone	02 9999 9999
Main Fax	02 9999 9999

There are more details below (press the Page Down key)

Contact Title	Mr
Contact First Name:	<Your First Name>
Contact Surname:	<Your Surname>
Contact Phone:	02 9999 9999
Contact Fax:	02 9999 9999
Contact Mobile:	<Your Mobile>
Contact Email:	yourname@yourdomain
User Title:	Mr
User First Name:	<Admin First Name>
User Surname:	<Admin Surname>
User Phone:	02 9999 9999
User Fax:	02 9999 9999
User Email:	<Your Mobile>
Paid From:	23/01/1999
Paid To:	23/01/2002
Status:	ACTIVE

Next >>

<< Previous

Cancel

FIG. 3a

Portal Website Number & Web Site Details

There are more details above (press the Page Up key)

Description Header:
<Enter Your Main Page Header Details>

Description Header:
<Enter Your Main Page Wordings and Content>

Description Header:
<Enter Your Main Page Footer Text>

Netseek Logo Path:
x:\chamber\images\logo.jpg

Netseek Logo URL:
x:\chamber\

Client Logo Path:
x:\chamber\images\logo.jpg

Client Logo URL:
x:\chamber\

URL:
http://www.freshchamber.net.au/

Netseek URL:
x:\chamber\

Requires Menu ?
No

Requires Fax ?
No

Keywords:
<Enter Default Keywords Here if Applicable>

<< Previous

Cancel

Next >>

FIG. 3b

Content / Product / Service Pages	
Number Of Content Pages:	<input type="text" value="5"/>
Order Seed	<input type="text" value="50"/>
Category	Information <input type="button" value="v"/>
Product/Service Name	<input type="text" value="<Prod/Service>"/>
Image File Path	<input type="text"/>
Product/Service URL	<input type="text" value="<Enter External URL Here If Applicable>"/>
Product/Service Description	<input type="text" value="<Enter Product / Service Information Here>"/>
Product/Service Footer	<input type="text" value="<Enter Your Page Footer Here>"/>
Start Date	<input type="text" value="23/07/1999"/>
End Date	<input type="text" value="23/07/2002"/>
Status:	<input type="text" value="ACTIVE"/> <input type="button" value="v"/>
<div><input type="button" value="Previous"/> <input type="button" value="Cancel"/> <input type="button" value="Next >>"/></div>	

FIG. 4

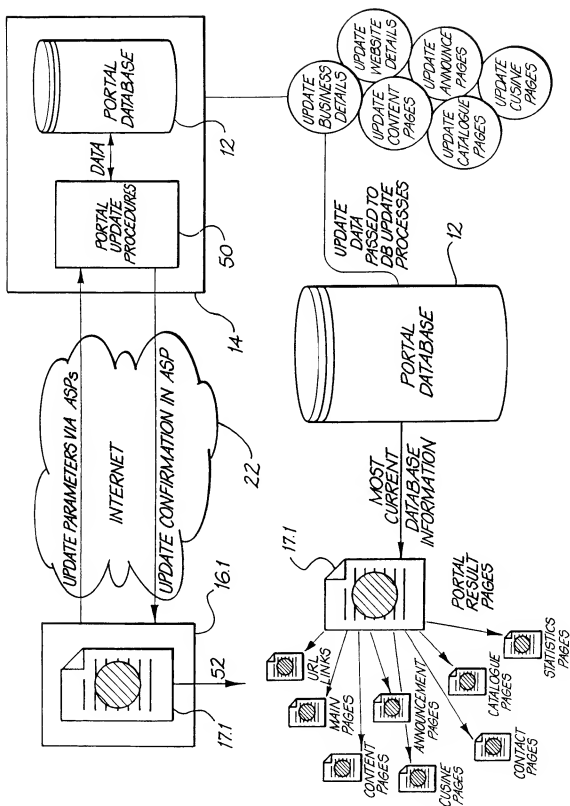


FIG. 5

Address <http://www.netseek.com.au/chamber/>

Select Website To Update

Business Details

Website Set-up

Home Page

Content Pages

User details

Your Statistics

Business Details

Update Your Business Details

Company Name

Address Line 1

Address Line 2

Suburb

Region

Postcode

State

Main Phone

Main Fax

Contact Name

Contact Phone

Contact Fax

Contact Mobile

Contact Email

Enable Faxing ☐

Client URL

Save Cancel

Please fill in all your company details.

Please input your contact details for any input are correct, or Netseek will email you or fax you system notifications.

User Name Patricia Roberson - Chamber

Netseek User/Usage Agreement

Copyright © 1999 Netseek Pty Ltd. All rights reserved.

Patent Pending

FIG. 6



Address  http://www.netseek.com.au/Chamber	
	
<p><Your Business Name> <Your Street> <Your Street> <Suburb> <State> <PostCode></p>	
<p>Phone: 02 9999 9999 Fax: 02 9999 9999 URL: http://<Your Domain Name></p>	
<p>Category: <Your Category></p>	
<p>Contact Details: Mr <First Name> <Surname></p>	
<p>Direct Phone: 02 9999 9999 Direct Fax: 02 9999 9999 Mobile: 9999 999 999</p>	
<p>Email: yourname@yourdomain.com.au</p>	
<p><Your Main Page Header> <Enter Your Main Page Content Text Here></p>	
<p><Enter Your Main Page Footer Text Here></p>	
<p>© 2000 <Your Company Name> Copyright Information / Usage Disclaimer Powered By Netseek™ Patent Pending</p>	

FIG. 7

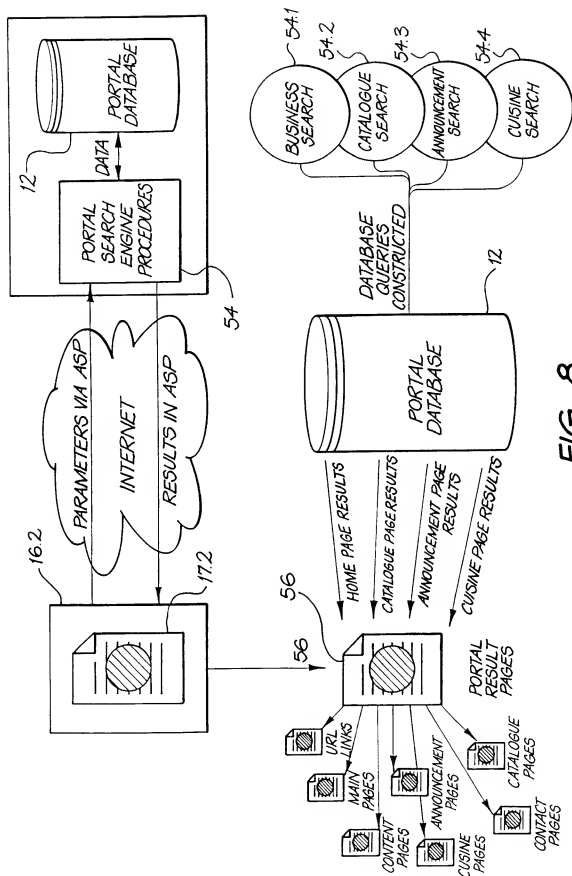


FIG. 8

Logo		Chamber Of Commerce	
Address http://www.netseek.com.au/chamber/			
<input type="checkbox"/> SEEK RESULTS Documents 1 to 4 of 4 matching your query			
Business / Web Site	Category/Description	Contact	Phone Suburb State
<Your Business Name Here>	<Your Category> - <Your Main Page Description>	<Firstname> <Surname>	02 9999 9999 Suburb NSW
<Your Business Name Here>	<Your Category> - <Your Main Page Description>	<Firstname> <Surname>	02 9999 9999 Suburb NSW
<Your Business Name Here>	<Your Category> - <Your Main Page Description>	<Firstname> <Surname>	02 9999 9999 Suburb NSW
<Your Business Name Here>	<Your Category> - <Your Main Page Description>	<Firstname> <Surname>	02 9999 9999 Suburb NSW
<Your Business Name Here>	<Your Category> - <Your Main Page Description>	<Firstname> <Surname>	02 9999 9999 Suburb NSW

Page 1 of 1

Search Now!

Advanced Seek Business Name:	<input type="text"/>
Type:	Automotive <input type="button" value="v"/>
Product/Service:	ALL <input type="button" value="v"/>
Results	

Search Businesses	<input type="text"/>
Keyword(s):	<input type="text"/>
Any Words <input type="button" value="v"/>	
Suburb:	<input type="text"/>
Region:	ALL <input type="button" value="v"/>
State:	NSW <input type="button" value="v"/>

FIG. 9

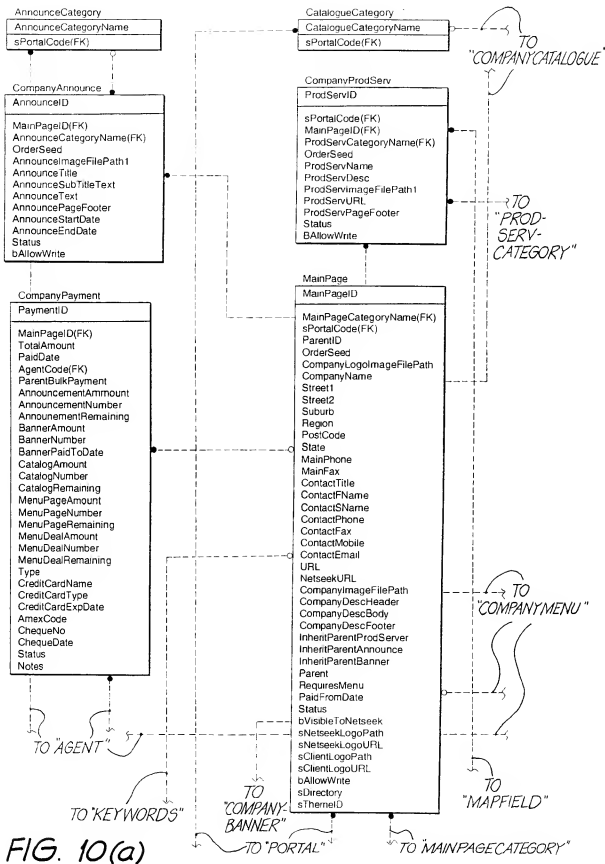
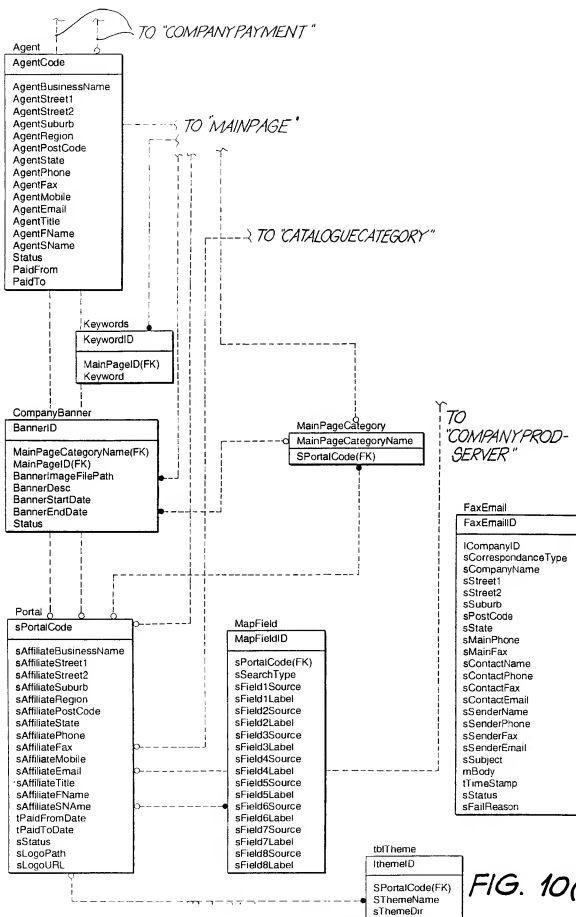
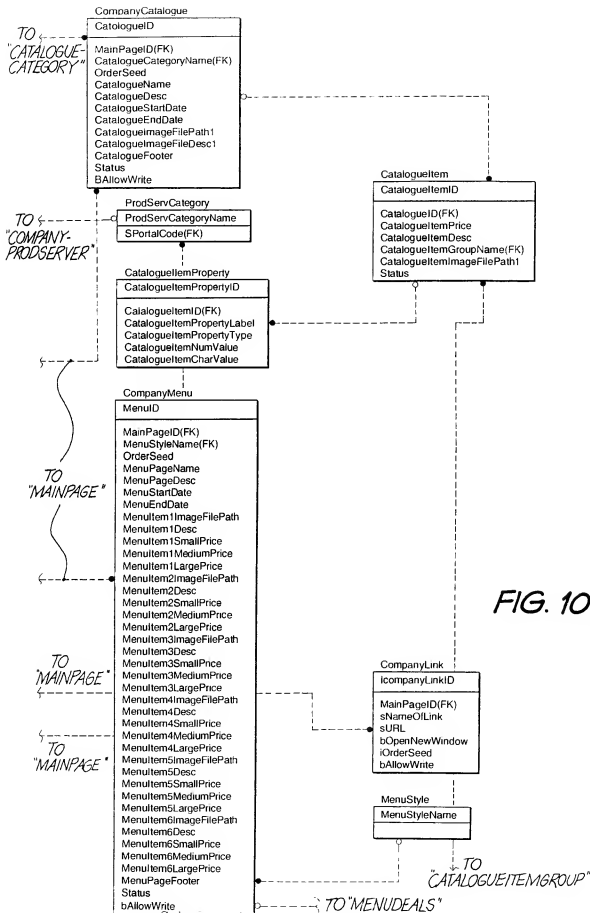


FIG. 10(a)





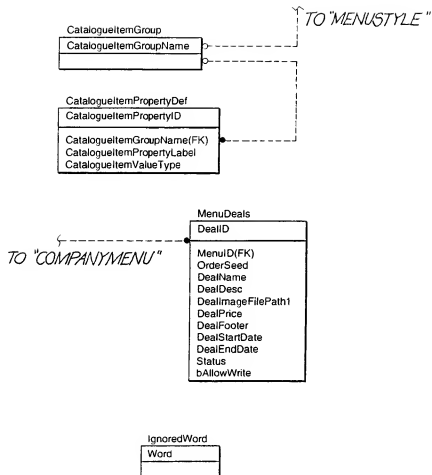


FIG. 10(d)

INTERNATIONAL SEARCH REPORT

International application No.

PCT/AU01/00393

A. CLASSIFICATION OF SUBJECT MATTERInt. Cl. ⁷² G06F 17/30, 17/60

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

WPAT, CORA, IEEE (web page, database)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5 859 972 (Subramaniam et al.) 12 January 1999 Abstract, Figures, 5A-5C, 8, 11, column 5, lines 20-42, column 8, lines 15-42 column 9, lines 9-18, column 11, lines 13-37	1-3, 12, 29, 30, 34, 35, 45, 60-65
Y		20, 45
X	"Querying Semantically Tagged Documents on the World-Wide Web" (Bar-Yossef et al.) Proceedings of the 4 th Workshop on Next Generation Information Technologies and Systems (NGITS) 1999 pp2-19	12
Y	Fig. 1, Section 3.1, Table 1, section 5.1 with respect to "new objects", "oid" object identifiers	20
Y	GB 2 324 896 (Mitel Corporation) 4 November 1998 Abstract, figures, claims	1, 20

☒ Further documents are listed in the continuation of Box C
 ☒ See patent family annex

* Special categories of cited documents:

- "A" document defining the general state of the art which is not considered to be of particular relevance
- "E" earlier application or patent but published on or after the international filing date
- "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- "O" document referring to an oral disclosure, use, exhibition or other means
- "P" document published prior to the international filing date but later than the priority date claimed

- "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
- "&" document member of the same patent family

Date of the actual completion of the international search

21 May 2001

Date of mailing of the international search report

31 May 2001 (31.05.01)

Name and mailing address of the ISA/AU

AUSTRALIAN PATENT OFFICE
PO BOX 200, WODEN ACT 2606, AUSTRALIA
E-mail address: pct@ipaustria.gov.au
Facsimile No. (02) 6285 3929

Authorized officer

DALE E. SIVER

Telephone No.: (02) 6283 2196

INTERNATIONAL SEARCH REPORT

International application No.

PCT/AU01/00393

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5 761 673 (Bookman et al.) 2 June 1998 Whole document	12,45
A	"Using Java and CORBA for Implementing Internet Databases" (Bouguettaya et al.) 15 th International Conference on Data Engineering 23-26 March 1999, pp 218-227 IEEE, Sections 1 and 2	71
A	US 5 884 309 (Vanechanos, Jr.) 16 March 1999 Abstract, figures, claims	12
A	"DataWeb: Customizable Database Publishing for the Web" (Miller et al.) IEEE Multimedia Vol. 4, No. 4, pp 14-21 Whole document	1

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.
PCT/AU01/00393

This Annex lists the known "A" publication level patent family members relating to the patent documents cited in the above-mentioned international search report. The Australian Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

Patent Document Cited in Search Report			Patent Family Member		
US	5859972	NO MEMBERS			
GB	2324896	CA 2231980	GB	9805270	US 5940834
US	5761673	NO MEMBERS			
US	5884309	NO MEMBERS			

END OF ANNEX